

## Lecture 27. Rayleigh Quotient, Inverse Iteration

In this lecture we present some classical eigenvalue algorithms. Individually, these tools are useful in certain circumstances—especially inverse iteration, which is the standard method for determining an eigenvector when the corresponding eigenvalue is known. Combined, they are the ingredients of the celebrated QR algorithm, described in the next two lectures.

### Restriction to Real Symmetric Matrices

Throughout numerical linear algebra, most algorithmic ideas are applicable either to general matrices or, with certain simplifications, to hermitian matrices. For the topics discussed in this and the next three lectures, this continues to be at least partly true, but some of the differences between the general and the hermitian cases are rather sizable. Therefore, in these four lectures, we simplify matters by considering only matrices that are real and symmetric. We also assume throughout that  $\|\cdot\| = \|\cdot\|_2$ .

Thus, for these four lectures:  $A = A^T \in \mathbb{R}^{m \times m}$ ,  $x \in \mathbb{R}^m$ ,  $x^* = x^T$ ,  $\|x\| = \sqrt{x^T x}$ . In particular, this means that  $A$  has real eigenvalues and a complete set of orthogonal eigenvectors. We use the following notation:

real eigenvalues:  $\lambda_1, \dots, \lambda_m$ ,

orthonormal eigenvectors:  $q_1, \dots, q_m$ .

The eigenvectors are presumed normalized by  $\|q_j\| = 1$ , and the ordering of the eigenvalues will be specified as necessary.

Most of the ideas to be described in the next few lectures pertain to Phase 2 of the two phases described in Lecture 25. This means that by the time we come to applying these ideas,  $A$  will be not just real and symmetric, but tridiagonal. This tridiagonal structure is occasionally of mathematical importance, for example in choosing shifts for the QR algorithm, and it is always of algorithmic importance, reducing many steps from  $O(m^3)$  to  $O(m)$  flops, as discussed at the end of the lecture.

## Rayleigh Quotient

The *Rayleigh quotient* of a vector  $x \in \mathbb{R}^m$  is the scalar

$$r(x) = \frac{x^T A x}{x^T x}. \quad (27.1)$$

Notice that if  $x$  is an eigenvector, then  $r(x) = \lambda$  is the corresponding eigenvalue. One way to motivate this formula is to ask: given  $x$ , what scalar  $\alpha$  “acts most like an eigenvalue” for  $x$  in the sense of minimizing  $\|Ax - \alpha x\|_2$ ? This is an  $m \times 1$  least squares problem of the form  $x\alpha \approx Ax$  ( $x$  is the matrix,  $\alpha$  is the unknown vector,  $Ax$  is the right-hand side). By writing the normal equations (11.9) for this system, we obtain the answer:  $\alpha = r(x)$ . Thus  $r(x)$  is a natural eigenvalue estimate to consider if  $x$  is close to, but not necessarily equal to, an eigenvector.

To make these ideas quantitative, it is fruitful to view  $x \in \mathbb{R}^m$  as a variable, so that  $r$  is a function  $\mathbb{R}^m \rightarrow \mathbb{R}$ . We are interested in the local behavior of  $r(x)$  when  $x$  is near an eigenvector. One way to approach this question is to calculate the partial derivatives of  $r(x)$  with respect to the coordinates  $x_j$ :

$$\begin{aligned} \frac{\partial r(x)}{\partial x_j} &= \frac{\frac{\partial}{\partial x_j}(x^T A x)}{x^T x} - \frac{(x^T A x) \frac{\partial}{\partial x_j}(x^T x)}{(x^T x)^2} \\ &= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x) 2x_j}{(x^T x)^2} = \frac{2}{x^T x} (Ax - r(x)x)_j. \end{aligned}$$

If we collect these partial derivatives into an  $m$ -vector, we find we have calculated the *gradient* of  $r(x)$ , denoted by  $\nabla r(x)$ . We have shown:

$$\nabla r(x) = \frac{2}{x^T x} (Ax - r(x)x). \quad (27.2)$$

From this formula we see that at an eigenvector  $x$  of  $A$ , the gradient of  $r(x)$  is the zero vector. Conversely, if  $\nabla r(x) = 0$  with  $x \neq 0$ , then  $x$  is an eigenvector and  $r(x)$  is the corresponding eigenvalue.

Geometrically speaking, the eigenvectors of  $A$  are the *stationary points* of the function  $r(x)$ , and the eigenvalues of  $A$  are the values of  $r(x)$  at these

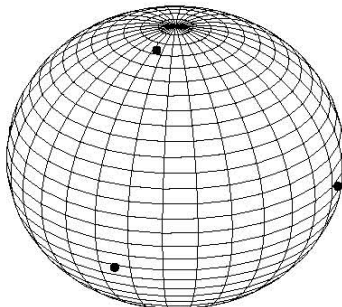


Figure 27.1. The Rayleigh quotient  $r(x)$  is a continuous function on the unit sphere  $\|x\| = 1$  in  $\mathbb{R}^m$ , and the stationary points of  $r(x)$  are the normalized eigenvectors of  $A$ . In this example with  $m = 3$ , there are three orthogonal stationary points (as well as their antipodes).

stationary points. Actually, since  $r(x)$  is independent of the scale of  $x$ , these stationary points lie along lines through the origin in  $\mathbb{R}^m$ . If we normalize by restricting attention to the unit sphere  $\|x\| = 1$ , they become isolated points (assuming that the eigenvalues of  $A$  are simple), as suggested in Figure 27.1.

Let  $q_J$  be one of the eigenvectors of  $A$ . From the fact that  $\nabla r(q_J) = 0$ , together with the smoothness of the function  $r(x)$  (everywhere except at the origin  $x = 0$ ), we derive an important consequence:

$$r(x) - r(q_J) = O(\|x - q_J\|^2) \quad \text{as } x \rightarrow q_J. \quad (27.3)$$

Thus the Rayleigh quotient is a *quadratically accurate* estimate of an eigenvalue. Herein lies its power.

A more explicit way to derive (27.3) is to expand  $x$  as a linear combination of the eigenvectors  $q_1, \dots, q_m$  of  $A$ . If  $x = \sum_{j=1}^m a_j q_j$ , then  $r(x) = \sum_{j=1}^m a_j^2 \lambda_j / \sum_{j=1}^m a_j^2$ . Thus  $r(x)$  is a weighted mean of the eigenvalues of  $A$ , with the weights equal to the squares of the coordinates of  $x$  in the eigenvector basis. Because of this squaring of the coordinates, it is not hard to see that if  $|a_j/a_J| \leq \epsilon$  for all  $j \neq J$ , then  $r(x) - r(q_J) = O(\epsilon^2)$ .

## Power Iteration

Now we switch tacks. Suppose  $v^{(0)}$  is a vector with  $\|v^{(0)}\| = 1$ . The following process, *power iteration*, was cited as a not especially good idea at the beginning of Lecture 25. It may be expected to produce a sequence  $v^{(i)}$  that converges to an eigenvector corresponding to the largest eigenvalue of  $A$ .

**Algorithm 27.1. Power Iteration**

$v^{(0)}$ = some vector with $\ v^{(0)}\  = 1$	
<b>for</b> $k = 1, 2, \dots$	
$w = Av^{(k-1)}$	apply $A$
$v^{(k)} = w/\ w\ $	normalize
$\lambda^{(k)} = (v^{(k)})^T Av^{(k)}$	Rayleigh quotient

In this and the algorithms to follow, we give no attention to termination conditions, describing the loop only by the suggestive expression “**for**  $k = 1, 2, \dots$ .” Of course, in practice, termination conditions are very important, and this is one of the points where top-quality software such as can be found in LAPACK or MATLAB is likely to be superior to a program an individual might write.

We can analyze power iteration easily. Write  $v^{(0)}$  as a linear combination of the orthonormal eigenvectors  $q_i$ :

$$v^{(0)} = a_1 q_1 + a_2 q_2 + \cdots + a_m q_m.$$

Since  $v^{(k)}$  is a multiple of  $A^k v^{(0)}$ , we have for some constants  $c_k$

$$\begin{aligned} v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \cdots + a_m \lambda_m^k q_m) \\ &= c_k \lambda_1^k (a_1 q_1 + a_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + a_m (\lambda_m/\lambda_1)^k q_m). \end{aligned} \quad (27.4)$$

From here we obtain the following conclusion.

**Theorem 27.1.** *Suppose  $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$  and  $q_1^T v^{(0)} \neq 0$ . Then the iterates of Algorithm 27.1 satisfy*

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \quad (27.5)$$

as  $k \rightarrow \infty$ . The  $\pm$  sign means that at each step  $k$ , one or the other choice of sign is to be taken, and then the indicated bound holds.

*Proof.* The first equation follows from (27.4), since  $a_1 = q_1^T v^{(0)} \neq 0$  by assumption. The second follows from this and (27.3). If  $\lambda_1 > 0$ , then the  $\pm$  signs are all  $+$  or all  $-$ , whereas if  $\lambda_1 < 0$ , they alternate.  $\square$

The  $\pm$  signs in (27.5) and in similar equations below are not very appealing. There is an elegant way to avoid these complications, which is to speak of convergence of subspaces, not vectors—to say that  $\langle v^{(k)} \rangle$  converges to  $\langle q_1 \rangle$ , for

example. However, we shall not do this, in order to avoid getting into the details of how convergence of subspaces can be made precise.

On its own, power iteration is of limited use, for several reasons. First, it can find only the eigenvector corresponding to the largest eigenvalue. Second, the convergence is linear, reducing the error only by a constant factor  $\approx |\lambda_2/\lambda_1|$  at each iteration. Finally, the quality of this factor depends on having a largest eigenvalue that is significantly larger than the others. If the largest two eigenvalues are close in magnitude, the convergence will be very slow.

Fortunately, there is a way to amplify the differences between eigenvalues.

## Inverse Iteration

For any  $\mu \in \mathbb{R}$  that is not an eigenvalue of  $A$ , the eigenvectors of  $(A - \mu I)^{-1}$  are the same as the eigenvectors of  $A$ , and the corresponding eigenvalues are  $\{(\lambda_j - \mu)^{-1}\}$ , where  $\{\lambda_j\}$  are the eigenvalues of  $A$ . This suggests an idea. Suppose  $\mu$  is close to an eigenvalue  $\lambda_J$  of  $A$ . Then  $(\lambda_J - \mu)^{-1}$  may be much larger than  $(\lambda_j - \mu)^{-1}$  for all  $j \neq J$ . Thus, if we apply power iteration to  $(A - \mu I)^{-1}$ , the process will converge rapidly to  $q_J$ . This idea is called *inverse iteration*.

### Algorithm 27.2. Inverse Iteration

$v^{(0)}$ = some vector with $\ v^{(0)}\  = 1$	
<b>for</b> $k = 1, 2, \dots$	
Solve $(A - \mu I)w = v^{(k-1)}$ for $w$	apply $(A - \mu I)^{-1}$
$v^{(k)} = w/\ w\ $	normalize
$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$	Rayleigh quotient

What if  $\mu$  is an eigenvalue of  $A$ , so that  $A - \mu I$  is singular? What if it is nearly an eigenvalue, so that  $A - \mu I$  is so ill-conditioned that an accurate solution of  $(A - \mu I)w = v^{(k-1)}$  cannot be expected? These apparent pitfalls of inverse iteration cause no trouble at all; see Exercise 27.5.

Like power iteration, inverse iteration exhibits only linear convergence. Unlike power iteration, however, we can choose the eigenvector that will be found by supplying an estimate  $\mu$  of the corresponding eigenvalue. Furthermore, the rate of linear convergence can be controlled, for it depends on the quality of  $\mu$ . If  $\mu$  is much closer to one eigenvalue of  $A$  than to the others, then the largest eigenvalue of  $(A - \mu I)^{-1}$  will be much larger than the rest. Using the same reasoning as with power iteration, we obtain the following theorem.

**Theorem 27.2.** *Suppose  $\lambda_J$  is the closest eigenvalue to  $\mu$  and  $\lambda_K$  is the second closest, that is,  $|\mu - \lambda_J| < |\mu - \lambda_K| \leq |\mu - \lambda_j|$  for each  $j \neq J$ . Furthermore,*

suppose  $q_J^T v^{(0)} \neq 0$ . Then the iterates of Algorithm 27.2 satisfy

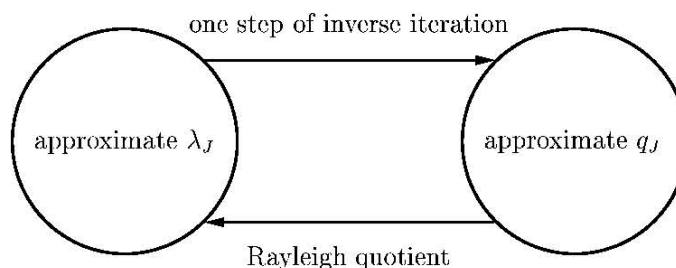
$$\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \quad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)$$

as  $k \rightarrow \infty$ , where the  $\pm$  sign has the same meaning as in Theorem 27.1.

Inverse iteration is one of the most valuable tools of numerical linear algebra, for it is the standard method of calculating one or more eigenvectors of a matrix if the eigenvalues are already known. In this case Algorithm 27.2 is applied as written, except that the calculation of the Rayleigh quotient is dispensed with.

### Rayleigh Quotient Iteration

So far in this lecture, we have presented one method for obtaining an eigenvalue estimate from an eigenvector estimate (the Rayleigh quotient), and another method for obtaining an eigenvector estimate from an eigenvalue estimate (inverse iteration). The possibility of combining these ideas is irresistible:



(The figure is oversimplified; to get from an approximate  $\lambda_J$  to an approximate  $q_J$  by a step of inverse iteration, one also needs a preliminary approximation to  $q_J$ .) The idea is to use continually improving eigenvalue estimates to increase the rate of convergence of inverse iteration at every step. This algorithm is called *Rayleigh quotient iteration*.

#### Algorithm 27.3. Rayleigh Quotient Iteration

$v^{(0)}$  = some vector with  $\|v^{(0)}\| = 1$   
 $\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$  = corresponding Rayleigh quotient  
**for**  $k = 1, 2, \dots$   
     Solve  $(A - \lambda^{(k-1)}I)w = v^{(k-1)}$  for  $w$       apply  $(A - \lambda^{(k-1)}I)^{-1}$   
      $v^{(k)} = w/\|w\|$       normalize  
      $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$       Rayleigh quotient

The convergence of this algorithm is spectacular: each iteration triples the number of digits of accuracy.

**Theorem 27.3.** *Rayleigh quotient iteration converges to an eigenvalue/eigenvector pair for all except a set of measure zero of starting vectors  $v^{(0)}$ . When it converges, the convergence is ultimately cubic in the sense that if  $\lambda_J$  is an eigenvalue of  $A$  and  $v^{(0)}$  is sufficiently close to the eigenvector  $q_J$ , then*

$$\|v^{(k+1)} - (\pm q_J)\| = O(\|v^{(k)} - (\pm q_J)\|^3) \quad (27.6)$$

and

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3) \quad (27.7)$$

as  $k \rightarrow \infty$ . The  $\pm$  signs are not necessarily the same on the two sides of (27.6).

*Proof.* We shall not prove the assertion about convergence for almost all starting vectors. Here, however, is a proof that if convergence occurs, it is ultimately cubic. For simplicity, we assume that the eigenvalue  $\lambda_J$  is simple. By (27.3), if  $\|v^{(k)} - q_J\| \leq \epsilon$  for sufficiently small  $\epsilon$ , then the Rayleigh quotient yields an eigenvalue estimate  $\lambda^{(k)}$  with  $|\lambda^{(k)} - \lambda_J| = O(\epsilon^2)$ . By the argument used to prove Theorem 27.2, if we now take one step of inverse iteration to obtain a new  $v^{(k+1)}$  from  $v^{(k)}$  and  $\lambda^{(k)}$ , then

$$\|v^{(k+1)} - q_J\| = O(|\lambda^{(k)} - \lambda_J| \|v^{(k)} - q_J\|) = O(\epsilon^3).$$

Moreover, the constants implicit in the  $O$  symbols are uniform throughout sufficiently small neighborhoods of  $\lambda_J$  and  $q_J$ . Thus we have convergence in the following pattern:

$$\begin{array}{ccc} \|v^{(k)} - (\pm q_J)\| & & |\lambda^{(k)} - \lambda_J| \\ \epsilon & \rightarrow & O(\epsilon^2) \\ \downarrow \swarrow & & \\ O(\epsilon^3) & \rightarrow & O(\epsilon^6) \\ \downarrow \swarrow & & \\ O(\epsilon^9) & \rightarrow & O(\epsilon^{18}) \\ \vdots & & \vdots \end{array}$$

The estimates (27.6)–(27.7) follow from the uniformity just mentioned.  $\square$

**Example 27.1.** Cubic convergence is so fast that we must give a numerical example. Consider the symmetric matrix

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix},$$

and let  $v^{(0)} = (1, 1, 1)^T / \sqrt{3}$  be the initial eigenvector estimate. When Rayleigh quotient iteration is applied to  $A$ , the following values  $\lambda^{(k)}$  are computed by the first three iterations:

$$\lambda^{(0)} = 5, \quad \lambda^{(1)} = 5.2131\dots, \quad \lambda^{(2)} = 5.214319743184\dots$$

The actual value of the eigenvalue corresponding to the eigenvector closest to  $v^{(0)}$  is  $\lambda = 5.214319743377$ . After only three iterations, Rayleigh quotient iteration has produced a result accurate to ten digits. Three more iterations would increase this figure to about 270 digits, if our machine precision were high enough.  $\square$

## Operation Counts

We close this lecture with a note on the amount of work required to execute each step of the three iterations we have described.

First, suppose  $A \in \mathbb{R}^{m \times m}$  is a full matrix. Then each step of power iteration involves a matrix-vector multiplication, requiring  $O(m^2)$  flops. Each step of inverse iteration involves the solution of a linear system, which might seem to require  $O(m^3)$  flops, but this figure reduces to  $O(m^2)$  if the matrix is processed in advance by LU or QR factorization or another method. In the case of Rayleigh quotient iteration, the matrix to be inverted changes at each step, and beating  $O(m^3)$  flops per step is not so straightforward.

These figures improve greatly if  $A$  is tridiagonal. Now, all three iterations require just  $O(m)$  flops per step. For the analogous iterations involving non-symmetric matrices, incidentally, we must deal with Hessenberg instead of tridiagonal structure, and this figure increases to  $O(m^2)$ .

## Exercises

**27.1.** Let  $A \in \mathbb{C}^{m \times m}$  be given, not necessarily hermitian. Show that a number  $z \in \mathbb{C}$  is a Rayleigh quotient of  $A$  if and only if it is a diagonal entry of  $Q^* A Q$  for some unitary matrix  $Q$ . Thus Rayleigh quotients are just diagonal entries of matrices, once you transform orthogonally to the right coordinate system.

**27.2.** Again let  $A \in \mathbb{C}^{m \times m}$  be arbitrary. The set of all Rayleigh quotients of  $A$ , corresponding to all nonzero vectors  $x \in \mathbb{C}^m$ , is known as the *field of values* or *numerical range* of  $A$ , a subset of the complex plane denoted by  $W(A)$ .

- (a) Show that  $W(A)$  contains the convex hull of the eigenvalues of  $A$ .
- (b) Show that if  $A$  is normal, then  $W(A)$  is equal to the convex hull of the eigenvalues of  $A$ .

**27.3.** Show that for a nonhermitian matrix  $A \in \mathbb{C}^{m \times m}$ , the Rayleigh quotient  $r(x)$  gives an eigenvalue estimate whose accuracy is generally linear,



not quadratic. Explain what convergence rate this suggests for the Rayleigh quotient iteration applied to nonhermitian matrices.

**27.4.** Every real symmetric square matrix can be orthogonally diagonalized, and the developments of this lecture are invariant under orthogonal changes of coordinates. Thus it would have been sufficient to carry out each derivation of this lecture under the assumption that  $A$  is a diagonal matrix with entries ordered by decreasing absolute value. Making this assumption, describe the form taken by (27.4), (27.5), and Algorithm 27.3.

**27.5.** As mentioned in the text, inverse iteration depends on the solution of a system of equations that may be exceedingly ill-conditioned, with condition number on the order of  $\epsilon_{\text{machine}}^{-1}$ . We know that it is impossible in general to solve ill-conditioned systems accurately. Is this not a fatal flaw in the algorithm?

Show as follows that the answer is no—that ill-conditioning is not a problem in inverse iteration. Suppose  $A$  is a real symmetric matrix with one eigenvalue much smaller than the others in absolute value (without loss of generality, we are taking  $\mu = 0$ ). Suppose  $v$  is a vector with components in the directions of all the eigenvectors  $q_1, \dots, q_m$  of  $A$ , and suppose  $Aw = v$  is solved backward stably, yielding a computed vector  $\tilde{w}$ . Making use of the calculation on p. 95, show that although  $\tilde{w}$  may be far from  $w$ ,  $\tilde{w}/\|\tilde{w}\|$  will not be far from  $w/\|w\|$ .

**27.6.** What happens to Figure 27.1 if two of the eigenvalues of  $A$  are equal?