
The QR Decomposition

- We have seen one major decomposition of a matrix which is $A = LU$ (and its variants) or more generally $\mathcal{P}A = LU$ for a permutation matrix \mathcal{P} . This was valid for a square matrix and aided us in solving the linear system $A\vec{x} = \vec{b}$.
- The QR decomposition is valid for rectangular matrices as well as square ones. We will first state the decomposition for square matrices.
- We will see that this decomposition can be used for solving $n \times n$ linear systems but is also useful in solving overdetermined systems such as those in linear least squares. The decomposition will be used in a general algorithm for finding all eigenvalues and eigenvectors of a matrix.

The QR Decomposition of a square matrix Let A be an $n \times n$ matrix with linearly independent columns. Then A can be uniquely written as $A^T A = QR$ where Q is orthogonal (unitary in general) and R is an upper triangular matrix with positive diagonal entries.

Outline of Proof

- The $n \times n$ matrix $A^T A$ is symmetric and positive definite and thus it can be written *uniquely* as $A = LL^T$ where L is lower triangular with positive diagonal entries.
- Let $Q = A(L^T)^{-1}$ and show that Q is an orthogonal matrix.
- Then $A = QL^T$ so set $R = L^T$ and we are done because L has positive diagonal entries.
- Uniqueness is demonstrated by assuming we have two such decompositions and getting a contradiction.

So all we have to do to show existence is demonstrate that $Q = A(L^T)^{-1}$ is an orthogonal matrix which means we have to demonstrate that $QQ^T = I$. Using the fact that $A^T A = LL^T$ we have

$$Q^T Q = (A(L^T)^{-1})^T (A(L^T)^{-1}) = L^{-1} A^T A (L^T)^{-1} = L^{-1} LL^T (L^T)^{-1} = I$$

All that remains is to verify uniqueness which should rely on the fact that the Cholesky decomposition is unique once we choose the sign of the diagonal entries. We assume there are two such decompositions and get a contradiction. Let

$$A = Q_1 R_1 \quad \text{and} \quad A = Q_2 R_2$$

where $Q_1^T Q_1 = I$ and $Q_2^T Q_2 = I$ and $R_1 \neq R_2$. Now writing $A^T A$ with each of these two decompositions gives

$$A^T A = (Q_1 R_1)^T (Q_1 R_1) = R_1^T Q_1^T Q_1 R_1 = R_1^T R_1$$

and

$$A^T A = (Q_2 R_2)^T (Q_2 R_2) = R_2^T Q_2^T Q_2 R_2 = R_2^T R_2$$

Thus

$$A^T A = R_1^T R_1 = R_2^T R_2$$

But this says that there are two different LL^T decompositions of $A^T A$ where each L has positive diagonal entries and thus we have a contradiction and the decomposition is unique. ■

- The proof of this theorem actually gives us a way to construct a QR decomposition of a matrix. We first form $A^T A$, do a Cholesky decomposition and thus have R and form $Q = AR^{-1}$. This can be done by hand, but is NOT a good approach computationally.
- The QR decomposition can be used to solve a linear system $A\vec{x} = \vec{b}$. We have

$$A\vec{x} = \vec{b} \implies QR\vec{x} = \vec{b} \implies Q^T QR\vec{x} = Q^T \vec{b} \implies R\vec{x} = Q^T \vec{b}$$

which is an upper triangular matrix. So once we have the factorization we have to do a matrix vector multiplication and solve an upper triangular system; both operations are $\mathcal{O}(n^2)$.

How can we obtain a QR decomposition?

- We saw two ways to obtain the LU factorization of a matrix. We can't take

the approach of equating elements of the matrix in the expression $A = QR$ because we need Q to be orthogonal so we turn to the other approach we used in finding the LU decomposition – we premultiplied by matrices which had the desired property (unit lower triangular in the LU) and were easy to multiply and invert.

- We will take an analogous approach here. Our goal is to find **orthogonal** matrices \mathcal{H}^i such that

$$\mathcal{H}^p \cdots \mathcal{H}^2 \mathcal{H}^1 A = R \implies A = (\mathcal{H}^1)^T (\mathcal{H}^2)^T \cdots (\mathcal{H}^p)^T R$$

where R is upper triangular. The matrices \mathcal{H}^i will play the same role that the Gauss transformation matrices played, i.e., they zero out entries before the main diagonal; however, we need \mathcal{H}^i to be orthogonal. We must also demonstrate that the product of two orthogonal matrices is orthogonal.

Let $\vec{u} \in \mathbb{R}^n$, $\|\vec{u}\|_2 \neq 0$. The $n \times n$ matrix

$$H = I - 2 \frac{\vec{u}\vec{u}^T}{\vec{u}^T\vec{u}}$$

is called a **Householder transformation** or a **Householder reflector** or an **elementary reflector**.

Exercise Verify that H is symmetric and orthogonal.

If the vector \vec{u} is chosen correctly, then H has the effect of zeroing out the first column of A below the diagonal when we premultiply A by it. If we know how to choose \vec{u} so that $H\vec{x} = (\hat{x}, 0, 0, \dots, 0)^T$ then we will know how to choose H to zero out any column of A below the diagonal.

Lemma Let $\vec{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ where $\vec{x} \neq \vec{0}$. Define

$$\alpha = \|\vec{x}\|_2 \beta \quad \text{where} \quad \beta = \begin{cases} 1 & \text{if } x_1 = 0 \\ x_1/|x_1| & \text{if } x_1 \neq 0. \end{cases}$$

Then the choice $\vec{u} = \vec{x} + \alpha \vec{e}_1$ in the definition of the Householder matrix gives

$$H\vec{x} = -\alpha \vec{e}_1,$$

where $\vec{e}_1 = (1, 0, 0, \dots, 0)^T \in \mathbb{R}^n$.

Example Let $\vec{x} = (0, -3, 4)^T$. Construct a Householder matrix H such that $H\vec{x} = c\vec{e}_1$.

From the lemma we set $\beta = 1$ and because $\|\vec{x}\|_2 = 5$ we have $\alpha = 5$. Then $\vec{u} = \vec{x} + 5\vec{e}_1 = (0, -3, 4)^T + (5, 0, 0)^T = (5, -3, 4)^T$ and $\vec{u}^T \vec{u} = 50$. Thus our Householder matrix is

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{2}{50} \begin{pmatrix} 25 & -15 & 20 \\ -15 & 9 & -12 \\ 20 & -12 & 16 \end{pmatrix} = \frac{1}{25} \begin{pmatrix} 0 & 15 & -20 \\ 15 & 16 & 12 \\ -20 & 12 & 9 \end{pmatrix}$$

As a check we compute $H\vec{x}$ to get $(-25/5, 0, 0)^T = -5(1, 0, 0)^T$ ■

So now we know how to compute an orthogonal matrix such that HA has zeros below the main diagonal in the first column. What do we do on the second and subsequent steps?

On the second step we want H^2H^1A to have zeros below the main diagonal in the second column but we don't want to modify the first column or row. So we simply choose

$$H^2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ & \hat{H}^2 & & \\ 0 & & & \end{pmatrix}$$

where \hat{H}^2 is chosen from our lemma such that $\hat{H}^2 \vec{x} = c\vec{e}_1$ where now $\vec{e}_1 \in \mathbb{R}^{n-1}$ and \vec{x} is the $n-1$ vector consisting of the $(2,2)$ entry of $H^1 A$ and the remainder of the column. For H^3 we choose a matrix where it is the 2×2 identity and an $(n-2) \times (n-2)$ matrix \hat{H}^3 chosen from our lemma. We make the analogous choices for each H^k as we proceed to transform A .

Thus we have

$$H^p \dots H^2 H^1 A = R \quad \text{where } R \text{ is upper triangular}$$

and because each H^k is symmetric and orthogonal we have

$$A = H^1 H^2 \dots H^p R$$

Now all we need to demonstrate is that the product of the H^k form an orthogonal matrix which we set to Q and we have $A = QR$. To see this, let U, V be orthogonal matrices so $UU^T = I$ and $VV^T = I$. Now the product UV is orthogonal because $(UV)(UV)^T = UVV^T U^T = UU^T = I$.

It is wasteful to explicitly compute and store each matrix H^k ; instead there is an efficient way to form the product of H^k (without forming it) and a matrix.

When should we use the QR factorization instead of an LU (or its variants) to solve $A\vec{x} = \vec{b}$?

One can demonstrate that the QR factorization requires $\mathcal{O}(n^3)$ just as the LU factorization does; however, the coefficient in front of the n^3 is considerably larger so that it is more expensive to compute. This is why the LU factorization is much more popular for solving linear systems. However, because the decomposition uses orthogonal matrices (and we will show that \mathcal{K}_2 of an orthogonal matrix is 1), it is often less sensitive to ill-conditioning than the LU factorization.

We can also perform a QR decomposition of an $m \times n$ matrix. In this case R is an $m \times n$ upper trapezoidal matrix and Q is an $m \times m$ orthogonal matrix.

Theoretical Results for the Eigenvalue Problem

Among problems in numerical linear algebra, the determination of the eigenvalues and eigenvectors of matrices is second in importance only to the solution of linear systems. Moreover, we need to have an understanding of their properties to fully understand our last decomposition, the Singular Value Decomposition as well as iterative methods for linear systems. Here we give some theoretical results relevant to the resolution of algebraic eigenvalue problems and defer the algorithms for calculating eigenvalues and eigenvectors to a later date. Recall our algebraic eigenvalue problem.

Eigenvalue problem Given an $n \times n$ matrix A , find a scalar λ (an eigenvalue) and a *nonzero* vector \vec{x} (an eigenvector) such that

$$A\vec{x} = \lambda\vec{x}$$

- Eigenvectors are sometimes called *principal directions*. Why? Typically when

we multiply a vector \vec{y} by a matrix A the resulting vector is different from \vec{y} in direction and magnitude. But when \vec{y} is an eigenvector of A the result $A\vec{y}$ is in the same direction as \vec{y} and is just scaled by λ , the corresponding eigenvalue.

- We note that if \vec{x} is an eigenvector of A then so is $c\vec{x}$ for any nonzero constant c . Thus eigenvectors will only be determined up to a multiplicative constant.
- If (λ, \vec{x}) is an eigenpair of A then $\vec{x} \in \mathcal{N}(A - \lambda I)$ because $A\vec{x} = \lambda\vec{x}$ implies $\vec{0} = A\vec{x} - \lambda I\vec{x} = (A - \lambda I)\vec{x}$.
- Clearly, λ is an eigenvalue of A if and only if the matrix $A - \lambda I$ is singular, i.e., if and only if the determinant $\det(A - \lambda I) = 0$.
- A has a zero eigenvalue if and only if it is singular.
- The equation $\det(A - \lambda I)$ is a polynomial of degree n and is called the **characteristic polynomial** of A .
- Because the characteristic polynomial is a polynomial of degree n we know that it has n roots, counted according to multiplicity; that is, some of the roots may be repeated. In addition, even if the coefficients of the polynomial

are real, the roots may occur in complex conjugate pairs. This means that a real matrix may have complex eigenvalues and thus complex eigenvectors.

Exercise Find the characteristic polynomial for the matrix A . Then determine its eigenvalues.

$$A = \begin{pmatrix} -2 & 2 \\ 2 & 1 \end{pmatrix}$$

- Because the characteristic polynomial is a polynomial of degree n we know that there is no formula for its solution when $n \geq 5$; moreover it is a nonlinear equation in λ . What does this tell us? It says that our algorithms for determining eigenvalues must be **iterative** in nature.
- General nonlinear equations can not be solved by direct methods but must be solved iteratively with methods such as Newton's method. However, finding the roots of the characteristic polynomial is NOT a good algorithm for finding eigenvalues for large n .

- The set of eigenvalues for a matrix A is called its **spectrum** and the largest eigenvalue in magnitude is its **spectral radius**, denoted $\rho(A)$. This says that if we plot the eigenvalues in the complex plane and draw a circle of radius ρ , then all eigenvalues are contained within that circle.
- If A is invertible with eigenpair $\{\lambda, \vec{x}\}$ then $\{1/\lambda, \vec{x}\}$ is an eigenpair of A^{-1} .
- If A has an eigenpair $\{\lambda, \vec{x}\}$ then $\{\lambda^k, \vec{x}\}$ is an eigenpair of A^k .

Exercise Let

$$A = \begin{pmatrix} -2 & 2 \\ 2 & 1 \end{pmatrix}$$

Determine the eigenvalues of A^2 , A^{-1} , $A + 4I$.

- We say that an eigenvalue λ has **algebraic multiplicity** (a.m.) m if it is repeated m times as a root of the characteristic polynomial. The sum of the algebraic multiplicities of the distinct eigenvalues of an $n \times n$ matrix is equal to n .

- We say that an eigenvalue λ has **geometric multiplicity** (g.m.) p if it has p *linearly independent* eigenvectors associated with it. The geometric multiplicity of an eigenvalue must be less than or equal to the eigenvalue's algebraic multiplicity; **g.m.** \leq **a.m.** The sum of the geometric multiplicities must be $\leq n$.
- If the geometric multiplicity of an eigenvalue of A is less than its algebraic multiplicity we call the eigenvalue and the matrix A **defective**. If A has n linearly independent eigenvectors then it is called **nondefective**; thus the eigenvectors of a nondefective $n \times n$ matrix form a basis for \mathbb{R}^n .

We know that if the a.m. of an eigenvalue is $m > 1$ then it may have $1, \dots, m$ linearly independent eigenvectors corresponding to it. But what do we know about eigenvectors corresponding to different eigenvalues?

Eigenvectors corresponding to distinct eigenvalues are linearly independent.

Why is this true? To understand this result we look at what would happen if two

eigenvectors corresponding to different eigenvalues were linearly dependent. Let $\{\lambda_1, \vec{v}_1\}$ and $\{\lambda_2, \vec{v}_2\}$ be eigenpairs of A where we assume $\lambda_1 \neq \lambda_2$. Because \vec{v}_1 and \vec{v}_2 are linearly dependent there are nonzero c_1, c_2 such that $c_1\vec{v}_1 + c_2\vec{v}_2 = \vec{0}$. Now multiplying this equation by A gives

$$c_1A\vec{v}_1 + c_2A\vec{v}_2 = \vec{0} \implies c_1\lambda_1\vec{v}_1 + c_2\lambda_2\vec{v}_2 = \vec{0}$$

Multiplying our original equation by λ_1 and subtracting gives

$$c_1\lambda_1\vec{v}_1 + c_2\lambda_2\vec{v}_2 - \lambda_1c_1\vec{v}_1 - \lambda_1c_2\vec{v}_2 = \vec{0} \implies c_2(\lambda_2 - \lambda_1)\vec{v}_2 = 0 \implies c_2 = 0$$

because $\lambda_2 - \lambda_1 \neq 0$ and $\vec{v}_2 \neq \vec{0}$ by assumption. If $c_2 = 0$ then $c_1\vec{v}_1 = 0$ and because $\vec{v}_1 \neq \vec{0}$, $c_1 = 0$ and we get our contradiction.

Example Consider the matrix

$$A = \begin{pmatrix} 3 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

Determine the eigenvalues of A and the algebraic and geometric multiplicities of each. Determine eigenvectors.

The characteristic polynomial is given $(3 - \lambda)^2(2 - \lambda)^2$ and the eigenvalues are 2 and 3, each having algebraic multiplicity 2. Also

$$A - 2I = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad A - 3I = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

so that $\mathcal{N}(A - 2I) = \text{span} \{(0 \ 0 \ 1 \ 0)^T, (0 \ 0 \ 0 \ 0 \ 1)^T\}$ and $\mathcal{N}(A - 3I) = \text{span}\{(1 \ 0 \ 0 \ 0)^T\}$. We conclude that the geometric multiplicity of the eigenvalue 2 is 2, and that of the eigenvalue 3 is 1. Thus, the eigenvalue 3 is defective, the eigenvalue 2 is nondefective, and the matrix A is defective.

- In general, if an eigenvalue λ of a matrix is known, then a corresponding eigenvector \vec{x} can be determined by solving for any particular solution of the singular system $(A - \lambda I)\vec{x} = \vec{0}$. A basis for this null space gives a set of linearly independent eigenvectors. Once again, this is not a procedure that is implemented on a computer.
- If an eigenvector \vec{x} of a matrix A is known then the corresponding eigenvalue

may be determined from the **Rayleigh quotient**

$$\lambda = \frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}.$$

which is found from the equation $\vec{x}^T A \vec{x} = \lambda \vec{x}^T \vec{x}$, i.e., multiplying the eigenvalue equation by \vec{x}^T .

Can we characterize the eigenvalues of specific classes of matrices?

It would be great if you could guarantee that certain matrices always have real eigenvalues, then we would know that we wouldn't have to use complex arithmetic. We summarize below some results for special classes of matrices.

If A is a symmetric matrix, then it has real eigenvalues.

If A is symmetric positive definite, then its eigenvalues are real and > 0 .

If A is orthogonal then its eigenvalues have magnitude 1.

If A is diagonal, then its eigenvalues are its diagonal entries.

If A is upper or lower triangular, then its eigenvalues are its diagonal entries.

Exercise Demonstrate that the eigenvalues of a symmetric, positive semi-definite matrix are ≥ 0 .

Because we can “read off” the eigenvalues of a diagonal or triangular matrix, then we might ask if there is some way we can transform our eigenvalue problem into an equivalent triangular or diagonal one. First we have to determine what types of transformations preserve eigenvalues.

Two matrices A, B are **similar** if there is an invertible matrix P such that

$$A = P^{-1}BP$$

Lemma Let A and B be similar and let (λ, \vec{x}) be an eigenpair of A then $(\lambda, P\vec{x})$ is an eigenpair of B .

This can be easily shown by noting that if $A\vec{x} = \lambda\vec{x}$ then

$$P^{-1}BP\vec{x} = \lambda\vec{x} \implies BP\vec{x} = \lambda P\vec{x} \implies B\vec{y} = \lambda\vec{y} \quad \text{where } \vec{y} = P\vec{x}$$

Now we want to know if we can make our matrix A similar to a triangular or diagonal matrix so we immediately know the eigenvalues. Theoretically this is possible as the next two results tell us.

Schur's Theorem. Let A be a given $n \times n$ matrix. Then there exists an $n \times n$ orthogonal matrix Q such that

$$Q^T A Q = U ,$$

where U is an upper triangular matrix whose diagonal entries are the eigenvalues of A . Furthermore, Q can be chosen so that the eigenvalues of A appear in any order along the diagonal of U . (Equivalently $A = Q U Q^T$)

Lemma. Let A be an $n \times n$ matrix. Then there exists an $n \times n$ invertible matrix P such that

$$P^{-1}AP = \Lambda,$$

if and only if A is **nondefective**. Here Λ is a diagonal matrix whose diagonal entries are the eigenvalues of A . Furthermore, P can be chosen so that the eigenvalues of A appear in any order along the diagonal of Λ . (Equivalently $A = P\Lambda P^{-1}$).

How can we find the matrix that makes a nondefective matrix A similar to a diagonal matrix (i.e., diagonalizes it)? Unfortunately, it turns out that if A is nondefective then **the columns of P are the eigenvectors of A** . But remember if we have the eigenvectors of a matrix, then we can use the Rayleigh quotient to get the corresponding eigenvalues so computationally this doesn't help us but theoretically it does.

Proof of Lemma Assume first that A is nondefective so it has a complete set of n linearly independent eigenvectors say \vec{v}_i so that $A\vec{v}_i = \lambda_i\vec{v}_i$. Let the columns of

P be the eigenvectors of A ; we know that P is invertible because it has linearly independent columns. Moreover

$$\begin{aligned} P^{-1}AP &= P^{-1}(A\vec{v}_1 \quad A\vec{v}_2 \quad \cdots \quad A\vec{v}_n) = P^{-1}(\lambda_1\vec{v}_1 \quad \lambda_2\vec{v}_2 \quad \cdots \quad \lambda_n\vec{v}_n) \\ &\implies P^{-1}AP = (\lambda_1P^{-1}\vec{v}_1 \quad \lambda_2P^{-1}\vec{v}_2 \quad \cdots \quad \lambda_nP^{-1}\vec{v}_n) = \Lambda \end{aligned}$$

where Λ is an $n \times n$ diagonal matrix containing the eigenvalues λ_i of A because $P^{-1}\vec{v}_i$ is the i th column of the identity matrix. Conversely if there exists an invertible P such that $P^{-1}AP = D$ where D is a diagonal matrix then $AP = PD$. If the diagonal entries of D are d_i and the columns of P are denoted \vec{p}_i then $AP = PD$ implies $A\vec{p}_i = d_i\vec{p}_i$ and the d_i are thus eigenvalues of A corresponding to the eigenvector \vec{p}_i . ■

What type of matrices have a complete set of linearly independent eigenvectors, i.e., are nondefective?

- Every diagonal matrix is nondefective because we can just choose \vec{e}_i as its eigenvectors.
- We know that if A has **distinct** eigenvalues it is nondefective, because eigen-

vectors corresponding to different eigenvalues are linearly independent.

- If A has repeated eigenvalues then their geometric multiplicities must equal their algebraic multiplicities in order for A to be nondefective.
- A **symmetric** matrix is guaranteed to be nondefective. This makes symmetric matrices especially nice because they have real eigenvalues and have a complete set of linearly independent eigenvectors. In fact, the following theorem gives a stronger statement about a symmetric matrix.

A symmetric $n \times n$ matrix has n orthonormal eigenvectors and its eigenvalues are real. Thus there exists an orthogonal matrix Q such that $Q^T A Q = \Lambda$ which implies $A = Q \Lambda Q^T$.

Recall that a set of vectors $\{\vec{v}_i\}_{i=1}^m$ are *orthonormal* if they are orthogonal and each has Euclidean length 1; i.e., $\vec{v}_i \cdot \vec{v}_j = 0$ for $i \neq j$ and $= 1$ for $i = j$. Recall that there is a procedure, called the Gram Schmidt Method which takes a set of linearly independent vectors and turns them into an orthonormal set of vectors.

If a matrix is nondefective then that means it has n linearly independent eigenvectors. We can then turn them into a set of orthonormal vectors using Gram Schmidt, but will they still be eigenvectors? The answer is a resounding “no”. Why?

Exercise We have seen that every diagonal matrix is nondefective. Is every upper triangular matrix nondefective? Prove or give a counterexample.

The spectral radius and the two-matrix norm

Recall that when we defined our induced matrix norm we gave a result which told us how to compute the matrix norms induced by the infinity and one vector norms but we did not have a means for calculating the matrix norm induced by the standard Euclidean vector norm, i.e., $\|A\|_2$. The following result gives us a way to calculate it.

Let A be an $n \times n$ matrix. Then

$$\|A\|_2 = \max_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} = \sqrt{\rho(A^T A)}$$

Recall that $A^T A$ is a symmetric positive semi-definite matrix so its eigenvalues are real and ≥ 0 so the expression makes sense.

If A is a symmetric matrix we can simplify this expression further because $A = A^T$ implies

$$\sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \rho(A)$$

This means that for a symmetric matrix we have the following result.

Let A be an $n \times n$ **symmetric** invertible matrix with eigenvalues ordered $0 < |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_n|$. Then

$$\|A\|_2 = \rho(A) \quad \text{and} \quad \mathcal{K}_2(A) = \frac{|\lambda_n|}{|\lambda_1|}$$

Exercise Show that if A is an $n \times n$ **symmetric** invertible matrix with eigenvalues ordered $0 < |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_n|$ and $\|A\|_2 = \rho(A)$ then $\mathcal{K}_2(A) = \frac{|\lambda_n|}{|\lambda_1|}$.

The Singular Value Decomposition

We saw that **symmetric** matrices were special in that they have real eigenvalues and a complete set of orthonormal eigenvectors; this told us that there is an **orthogonal** matrix Q such that

$$Q^T A Q = \Lambda \implies A = Q \Lambda Q^T .$$

We can view this as a decomposition of our matrix A into the product $Q \Lambda Q^T$ where Λ is an $n \times n$ diagonal matrix with the eigenvalues of A on its diagonal. Note that in this decomposition for symmetric matrices we only need one orthogonal matrix Q .

However, this decomposition is only guaranteed for symmetric matrices. What kind of decomposition is guaranteed for nonsymmetric matrices? The answer is given by the Singular Value Decomposition (SVD) Theorem. In this theorem we will see that, in general, two orthogonal matrices are needed to accomplish the decomposition of A .

The SVD is our third decomposition of a matrix and it holds for a general $m \times n$ matrix. In the past twenty-five years researchers have come to realize that it is extremely valuable in many applications. In the lab we will see how it can be used in image compression.

We will see that the SVD of A gives us information about the four fundamental spaces associated with A . In addition it will provide us with information on the relative importance of the columns of A . Remember that it holds for rectangular as well as square matrices.

The Singular Value Decomposition Theorem (SVD). Let A be an $m \times n$ matrix. Then A can be factored as

$$A = U\Sigma V^T$$

where

- U is an $m \times m$ orthogonal matrix
- Σ is an $m \times n$ diagonal matrix ($\Sigma_{ij} = 0$ for $i \neq j$) with entries σ_i
- V is an $n \times n$ orthogonal matrix

The diagonal entries of Σ , σ_i , are called the **singular values** of A and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$ where $k = \min\{m, n\}$.

Note that our decomposition of a symmetric matrix into $A = Q\Lambda Q^T$ is a special case of this decomposition when $U = V$.

If A is square, then so is Σ but if A is rectangular, so is Σ . For example, the

following matrices illustrate some possible forms for Σ ;

$$\begin{pmatrix} 10 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \begin{pmatrix} 10 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 10 & 0 \\ 0 & 4 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 10 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 10 & 0 & 0 \\ 0 & 4 & 0 \end{pmatrix}$$

Exercise Show how the SVD can be used to solve a linear system $A\vec{x} = \vec{b}$. However, this is an extremely expensive approach and is not recommended.

The SVD will be used in the efficient solution of the linear least squares problem which will be done later in the course.

The SVD can also be used to calculate A^{-1} if A is square or the so-called **pseudo inverse** $A^\dagger = (A^T A)^{-1} A^T$ if A is rectangular.

Exercise Show that if A is square and invertible, then $A^{-1} = A^\dagger$.

Let's look at what the SVD decomposition tells us. To do this, we first form the matrices $A^T A$ and AA^T and use the SVD of A to get the following.

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T U^T U\Sigma V^T .$$

Now U is orthogonal and thus $U^T U = I_{m \times m}$ implies

$$A^T A = V\Sigma^T U^T U\Sigma V^T = V(\Sigma^T \Sigma)V^T .$$

This says that $A^T A$ is orthogonally similar to $\Sigma^T \Sigma$; here $\Sigma^T \Sigma$ is an $n \times n$ diagonal matrix with entries σ_i^2 . Recall that this means they have the same eigenvalues.

Now consider the matrix AA^T

$$AA^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U(\Sigma\Sigma^T)U^T$$

which says that AA^T is orthogonally similar to $\Sigma\Sigma^T$; here $\Sigma\Sigma^T$ is an $m \times m$ diagonal matrix with entries σ_i^2 .

Now we partition the $n \times n$ matrix V as $V = (V_1|V_2)$ where V_1 is $n \times p$ and V_2 is $n \times (n - p)$ where p is the index of the last nonzero singular value σ_i ; i.e., $\sigma_{p+1} = \sigma_{p+2} = \dots = \sigma_k = 0$ where $k = \min\{m, n\}$. Likewise partition $U = (U_1|U_2)$

We now want to use these results to interpret the meaning of each matrix in the SVD.

1. The columns of V are the orthonormal eigenvectors of $A^T A$ and are called the **right singular vectors** of A because $AV = U\Sigma$.

Clearly $A^T A$ is symmetric and thus has a complete set of orthonormal eigenvectors. Also $A^T A = V(\Sigma^T \Sigma)V^T$ which implies $V^T(A^T A)V = \Sigma^T \Sigma$ so V is the matrix which diagonalizes $A^T A$ and thus the columns of V are its orthonormal eigenvectors.

2. The columns of U are the orthonormal eigenvectors of AA^T and are called the **left singular vectors** of A because $U^T A = \Sigma V^T$.

Clearly AA^T is symmetric and thus has a complete set of orthonormal eigenvectors. From above $AA^T = U(\Sigma \Sigma^T)U^T$ which implies $U^T(AA^T)U = \Sigma \Sigma^T$ so U is the matrix which diagonalizes AA^T and each of the columns of U is an orthonormal eigenvector of AA^T .

3. The singular values of A , σ_i , are the positive square roots of the eigenvalues of $A^T A$ and AA^T and are ≥ 0 .

V is the matrix which diagonalizes $A^T A V^T (A^T A) V = \Sigma^T \Sigma$ where $A^T A$ has eigenvalues σ^2 . U is the matrix which diagonalizes $AA^T U^T (AA^T) U = \Sigma \Sigma^T$ where AA^T has eigenvalues σ^2 .

4. An orthonormal basis for the null space of A , $\mathcal{N}(A)$, is given in V_2 .

To see this note that $A = U\Sigma V^T$ implies $AV = U\Sigma$ implies $(AV_1|AV_2) = U\Sigma$. Because the last $(p + 1)$ through n columns of V correspond to the diagonal entries of Σ which are zero, then $AV_2 = 0$ and the columns of V_2 are in $\mathcal{N}(A)$ and are orthonormal because V is orthogonal.

5. An orthonormal basis for the row space of A , $\mathcal{R}(A^T)$, is given in V_1 .

The first p columns of V denoted by V_1 are orthonormal to the columns of V_2 and form a basis for $\mathcal{R}(A^T)$ because $\mathcal{R}(A^T)$ is the orthogonal complement of $\mathcal{N}(A)$.

6. An orthonormal basis for the range of A , $\mathcal{R}(A)$, is given in U_1 .

As above, the SVD implies $AV = U\Sigma$ implies $A(V_1|V_2) = (U_1|U_2)\Sigma$. The first p columns of U denoted by U_1 form a basis for the range of A .

7. An orthonormal basis for the left nullspace of A , $\mathcal{N}(A^T)$, is given in U_2 .

The columns of U denoted by U_2 are orthogonal to U_1 and form a basis for the orthogonal complement $\mathcal{R}(A)^\perp = \mathcal{N}(A^T)$.

8. The rank of A is given by the number of nonzero singular values in Σ .

If we multiply a matrix by an orthogonal matrix it does not change its rank.

9. Let A be square and invertible then $\mathcal{K}_2(A)$ is defined to be σ_1/σ_k where σ_k is the smallest singular value > 0 . This definition can also be used for general matrices.

Recall that $\|B\|_2 = \sqrt{\rho(B^T B)}$ so $(\mathcal{K}_2(B))^2 = \|B\|_2^2 \|B^{-1}\|_2^2$ and thus $(\mathcal{K}_2(B))^2 = \rho(B^T B)\rho((B^T B)^{-1})$

Example Let

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 1 & 4 & 0 \\ 5 & 3 & 1 \end{pmatrix}$$

and where its SVD is given by

$$A = \begin{pmatrix} -.4256 & .6968 & -.5774 \\ -.3906 & -.7170 & -.5774 \\ -.8162 & -.0202 & .5774 \end{pmatrix} \begin{pmatrix} 7.247 & 0 & 0 \\ 0 & 4.1807 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -.8520 & .4710 & -.2287 \\ -.4948 & -.8671 & .0572 \\ -.1714 & .1618 & .9718 \end{pmatrix}^T$$

Find the singular values of A , the rank of A , a basis for the four fundamental spaces and their dimension using the SVD.

The singular values of A are the diagonal entries of Σ so here they are $\{7.2472, 4.1807, 0\}$. Clearly the rank of A is 2 because it has 2 nonzero singular values which of course means that A is singular. To find the basis for $\mathcal{R}(A)$ we partition U where U_1 denotes the first 2 columns of U and U_2 the last column. Then the columns of U_1 $\{(-.4256, -.3906, -.8162)^T, (.6968, -.7170, -.0202)^T\}$ form a basis for $\mathcal{R}(A)$ with dimension 2 and the column of U_2 form a basis for its orthogonal complement, $\mathcal{N}(A^T)$ (dimension 1). We partition V in the same way so its first two columns $\{(-.8520, -.4949, -.1714)^T, (.4710, -.8670, .1618)^T\}$ form a basis for $\mathcal{N}(A)^\perp = \mathcal{R}(A^T)$ (dimension 2) and the last column $(-.2287, .0572, .9718)^T$ forms a basis for $\mathcal{N}(A)$ (dimension 1). Note that in the decomposition we have given V , not V^T .

Exercise If A is symmetric, relate its eigenvalues and its singular values.

The fact that the eigenvectors of $A^T A$ are the columns of V and the eigenvectors of AA^T are the eigenvectors of U suggest that a possible means of obtaining the SVD would be to calculate the eigenvectors of $A^T A$ and AA^T , but it turns out that that approach does not lead to a robust algorithm.

The SVD provides another important piece of information about A concerning the relative importance of its columns. Suppose that we have n vectors in \mathbb{R}^m and we form an $m \times n$ matrix A with $\text{rank} \leq m$. Note that we can rewrite the SVD of A in an alternate form to get

$$A = \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_k \vec{u}_k \vec{v}_k^T$$

where \vec{u}_i represents the i th column of U and \vec{v}_i the i th column of V and k is the last nonzero singular value of A . Recall that the singular values are ordered from largest to smallest and that the vectors \vec{u}_i, \vec{v}_j have length 1. Thus the largest contribution to A occurs in the first term, the second largest in the next term, etc. Now suppose we want to approximate our matrix A by a matrix of rank

$\ell < \text{rank}(A)$; call it A_ℓ . Then we take

$$A_\ell = \sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^T .$$

Example Consider the following rank 3 matrix and its SVD. Use this to find rank 1, rank 2, and rank 3 approximations of A .

$$A = \begin{pmatrix} 2. & 1. & 1. \\ 10. & 3. & 4. \\ 8. & 1. & 4. \\ 6. & 0. & 8. \\ 4. & 6. & 8. \end{pmatrix}$$

where the $U\Sigma V^T$ is

$$\begin{pmatrix} -0.122 & 0.045 & 0.141 & 0.268 & -0.944 \\ -0.552 & 0.468 & 0.415 & 0.469 & 0.289 \\ -0.448 & 0.400 & -0.057 & -0.783 & -0.154 \\ -0.486 & -0.125 & -0.821 & 0.272 & 0.012 \\ -0.493 & -0.777 & 0.361 & -0.149 & 0.038 \end{pmatrix} \begin{pmatrix} 19.303 & 0.000 & 0.000 \\ 0.000 & 6.204 & 0.000 \\ 0.000 & 0.000 & 4.111 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{pmatrix} \begin{pmatrix} -0.738 & 0.664 & 0.121 \\ -0.269 & -0.453 & 0.850 \\ -0.619 & -0.595 & -0.512 \end{pmatrix}^T$$

To determine a rank 1 approximation we have

$$A_1 = 19.3 \begin{pmatrix} -.122 \\ -.552 \\ -.448 \\ -.486 \\ -.493 \end{pmatrix} \begin{pmatrix} -.7238 & -.269 & -.619 \end{pmatrix} = \begin{pmatrix} 1.743 & 0.635 & 1.464 \\ 7.864 & 2.864 & 6.603 \\ 6.379 & 2.323 & 5.356 \\ 6.920 & 2.520 & 5.811 \\ 7.021 & 2.557 & 5.895 \end{pmatrix}$$

In the same way a rank two approximation can be found to be

$$\begin{pmatrix} 1.930 & 0.508 & 1.297 \\ 9.794 & 1.548 & 4.875 \\ 8.028 & 1.199 & 3.880 \\ 6.407 & 2.870 & 6.270 \\ 3.821 & 4.738 & 8.760 \end{pmatrix}$$

The rank 3 approximation is A itself because A is rank three.

Model Order Reduction – An Application of the SVD

Suppose that you have a set of differential equations which model some physical, chemical, biological, etc. phenomena and when you discretize your model you get a large system of either linear or nonlinear equations. Further suppose that you either need a solution in real time or perhaps you need to do a parameter study which involves solving your system for a large range of parameters.

Suppose we have in hand a code that solves this problem but either you can not get the solution in real time or the number of studies you need to complete are prohibitively time consuming.

In reduced order modeling one generates a set of **snapshots** by

- solving the discretized PDEs for one or more specific sets of values of the parameters
- and/or sampling the approximate solutions at several instants in time

We want to use this set of snapshots as a **reduced basis** in which to seek our solution.

However, the snapshots contain a lot of redundant information. How can we distill these snapshots to remove the redundancy? The answer is found with the aid of the SVD.

Our hope is that we can use a small number of these basis vectors and seek the solution to the differential equation as a linear combination of these basis vectors. If the number of basis vectors is small, e.g., < 20 then we will have a small dense system to solve instead of a very large banded or sparse system.

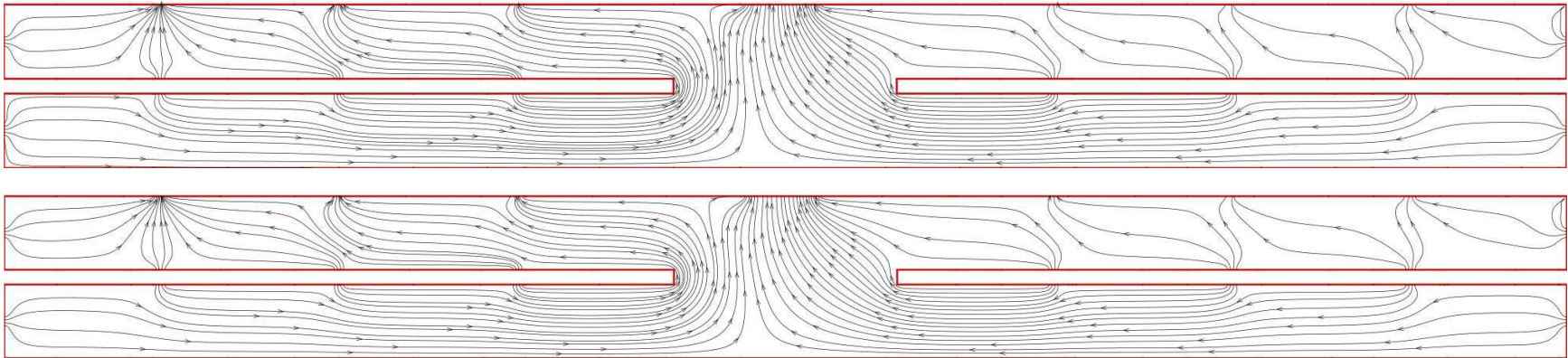
Proper Orthogonal Decomposition

The technique of using the SVD to remove the redundancy in the snapshots is called Proper Orthogonal Decomposition (POD). There are other methods such as clustering to do this too.

The steps of POD can be briefly described as follows:

- we start with a snapshot set $\{\vec{u}_j\}_{j=1}^N$
- we form the snapshot matrix A whose columns are the snapshots, i.e., $A = (\vec{u}_1 \ \vec{u}_2 \ \cdots \ \vec{u}_N)$
- we compute $A = U\Sigma V^T$, the SVD of A
- the K -dimensional POD basis is given by the first K left singular vectors of A , i.e., the first K columns of U

Recall that the singular vectors are associated with the singular values which occur in nonincreasing order so that we might take the first m vectors where m is chosen so that σ_m is less than some tolerance.



Flow in a prototype public building determined using a finite element code with $\approx 36,000$ of degrees of freedom (top) and with a reduced-order model with 8 degrees of freedom (bottom)

Algorithms for Finding Eigenvalues and Eigenvectors

Recall that we said that algorithms for determining eigenvalues and eigenvectors must be **iterative**.

In general, the method for finding all eigenvalues and eigenvectors is more expensive than solving a linear system.

In many problems (such as for stability analysis), we are only interested in finding a single eigenvalue such as the spectral radius.

We will begin by looking at techniques for finding a single eigenpair. Because symmetric matrices have real eigenvalues, we will first concentrate on that special case.

Before we look at the algorithms, we give a localization theorem for eigenvalues which could be used to get an initial guess for an algorithm.

Gerschgorin's Circle Theorem Let A be an $n \times n$ matrix and define the disks (i.e., circles) in the complex plane by

$$\mathcal{D}_i = \left\{ z : |a_{ii} - z| \leq \sum_{j \neq i} |a_{ij}| \right\} \quad i = 1, 2, \dots, n.$$

Then all the eigenvalues of A lie in the union of the disks $\cup_{i=1}^n \mathcal{D}_i$. Moreover, if k disks are disjoint then there are exactly k eigenvalues lying in the union of these k disks.

Exercise Apply Gerschgorin's theorem to the matrix

$$A = \begin{pmatrix} 2 & 2 & 2 \\ 2 & 4 & 1 \\ 1 & 1 & 10 \end{pmatrix}$$

The Power Method and its Variants

We now look at algorithms for determining a specific eigenvalue; for example the spectral radius of a matrix, the minimum eigenvalue (in magnitude) or the eigenvalue nearest a specific value.

The Power Method is an iterative method for find $\rho(A)$ which simply takes powers of the original matrix A . The assumption for the method to work is that A must have a **complete set of linearly independent eigenvectors**.

We will assume that A is symmetric so that it is guaranteed to have real eigenvalues and a complete set of linearly independent eigenvectors.

The method will give us an approximation to the eigenvector corresponding to $\rho(A)$ and we will use the Rayleigh Quotient or even a simpler method to obtain an approximation to the corresponding eigenvalue.

Power Method - Case 1

Let A be symmetric. Assume that the dominant eigenvalue is unique (not repeated) and the eigenvalues are ordered as follows

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

and the associated eigenvectors are denoted \vec{v}_i , $i = 1, \dots, n$.

Given \vec{x}^0 ; generate a sequence of iterates \vec{x}^k from

$$\vec{x}^k = \delta_k A \vec{x}^{k-1}$$

where δ_k is a suitably chosen scaling factor.

Then $\vec{x}^k \rightarrow C\vec{v}_1$ as $k \rightarrow \infty$ with rate $(|\lambda_2/\lambda_1|)^k$.

In this algorithm we are simply multiplying the previous iterate by the matrix A and scaling the result. We have to scale because otherwise the entries in the resulting vectors could become unbounded or approach zero. We now want to

see why this method converges to the eigenvector corresponding to the dominant eigenvalue and at what rate it converges.

Note that for each iteration we need to do a matrix times vector multiplication which is $\mathcal{O}(n^2)$.

To see why $x^k \rightarrow C\vec{v}_1$, we note that

$$\vec{x}^1 = \delta_1 A \vec{x}^0, \quad \vec{x}^2 = \delta_2 A \vec{x}^1 = \delta_1 \delta_2 A^2 \vec{x}^0$$

Continuing in this manner we see that

$$\vec{x}^k = \delta_k A \vec{x}^{k-1} = \delta_k \delta_{k-1} A^2 \vec{x}^{k-2} = \dots = \left(\prod_{i=1}^k \delta_i \right) A^k \vec{x}^0.$$

Because we have assumed that A has a complete set of linearly independent eigenvectors these vectors can be used as a basis for \mathbb{R}^n . Consequently there are constants c_i such that

$$x^0 = \sum_{i=1}^n c_i \vec{v}_i$$

Using this expression in our formula for x^k and the fact that $A^k \vec{v}_i = \lambda_i^k \vec{v}_i$ gives

$$\vec{x}^k = \left(\prod_{i=1}^k \delta_i \right) A^k \vec{x}^0 = \epsilon_k A^k \left[\sum_{i=1}^n c_i \vec{v}_i \right] = \epsilon_k \left[\sum_{i=1}^n c_i A^k \vec{v}_i \right] = \epsilon_k \left[\sum_{i=1}^n c_i \lambda_i^k \vec{v}_i \right].$$

where we have written the product of the constants $\prod_{i=1}^k \delta_i$ as ϵ_k for ease of exposition. We now factor out the dominant eigenvalue term λ_1^k to get

$$\vec{x}^k = \epsilon_k \lambda_1^k \left[c_1 \vec{v}_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \vec{v}_2 + c_3 \left(\frac{\lambda_3}{\lambda_1} \right)^k \vec{v}_3 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \vec{v}_n \right].$$

As $k \rightarrow \infty$ all the terms in the expression except the first approach zero because we have assumed that $\lambda_1 > \lambda_i$ for all $i \neq 1$. Now if we didn't scale, then the first term either approaches ∞ or 0 depending on whether $\lambda_1 > 1$ or $\lambda_1 < 1$.

As $k \rightarrow \infty$ the largest term in the expression (not counting the first) is a constant times $(\lambda_2/\lambda_1)^k$ because we have assumed the ordering $|\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$. Consequently the rate at which we have convergence is governed by $(|\lambda_2|/|\lambda_1|)^k$. This means, e.g., if $\lambda_2 = .5\lambda_1$ at the tenth iteration we would have $(\lambda_2/\lambda_1)^{10} = (0.5)^{10} \approx .000976$ but if the eigenvalues are clustered, e.g., $\lambda_2 = 0.95\lambda_1$ at the tenth iteration we would have $(\lambda_2/\lambda_1)^{10} = (.95)^{10} \approx 0.5987$ and the convergence would be very slow.

Example Consider the matrix

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

which has eigenvalues -6 , -3 and -1 and $\vec{v}_1 = k(1, -1, 1)^T$ for any scalar k . We apply the Power Method to find the eigenvector corresponding to the dominant eigenvalue and we scale using the infinity norm of the current iterate. The components of the iterate \vec{x}^k are given as well as the eigenvalue approximation found by using the Rayleigh Quotient. Note that the iteration is converging to a constant times $(1, -1, 1)^T$.

k	\vec{x}^k	λ_k
0	(1.00000 , .00000 ,.00000)	
1	-(1.00000,-.25000, .25000)	-4.0000
2	(1.00000,-.50000, .50000)	-5.0000
3	-(1.00000,- .70000,.70000)	-5.6667
4	(1.00000, -.83333,.83333)	-5.9091
5	-(1.00000,-.91176,.91176)	-5.9767
6	(1.00000, -.95455,.95455)	-5.9942
7	-(1.00000,- .97692,.97692)	-5.9985
8	(1.00000, -.98837,.98837)	-5.9996
9	-(1.00000, -.99416,.99416)	-5.9999
10	(1.00000, -.99708,.99708)	-6.0000

How do we compute the corresponding eigenvalues?

The obvious way, of course, is to use the Rayleigh quotient

$$\frac{(\vec{x}^k)^T A \vec{x}^k}{(\vec{x}^k)^T \vec{x}^k} = \frac{(\vec{x}^k)^T \vec{x}^{k+1}}{\delta_{k+1} (\vec{x}^k)^T \vec{x}^k}$$

Note that this last expression is how it should be implemented because we need to form $A\vec{x}^k$ anyway.

An alternate way to calculate the approximate eigenvalue is to take the ratio of components of successive iterates, e.g.,

$$\frac{(A\vec{x}^k)_\ell}{(\vec{x}^k)_\ell}$$

Example Let A be given by

$$A = \begin{pmatrix} -8.1 & 10.4 & 14.3 \\ 4.9 & -5. & -7.9 \\ -9.05 & 10.4 & 15.25 \end{pmatrix},$$

for which the spectrum of A is $\{1, .95, .2\}$; note that $k(1, -.5, 1)^T$ is an eigenvector corresponding to the eigenvalue $\lambda = 1$. Since $|\lambda_2/\lambda_1| = .95$, the iterates theoretically converge at a rate of $O(.95^k)$. The approximations λ_k to the eigenvalue λ_1 are computed using the Rayleigh quotient.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	λ_k	$\left \frac{\lambda_2}{\lambda_1}\right ^k$	$ \lambda_1 - \lambda_k $
0	1.00000	.00000	.00000			
1	.89503	-.54144	1.00000	-.27000E+01	.95000E+00	.37000E+01
2	.93435	-.53137	1.00000	.15406E+01	.90250E+00	.54064E+00
3	.95081	-.52437	1.00000	.12747E+01	.85737E+00	.27473E+00
4	.96079	-.51957	1.00000	.11956E+01	.81451E+00	.19558E+00
5	.96765	-.51617	1.00000	.11539E+01	.77378E+00	.15389E+00
6	.97267	-.51366	1.00000	.11264E+01	.73509E+00	.12645E+00
7	.97651	-.51175	1.00000	.11066E+01	.69834E+00	.10661E+00
8	.97953	-.51023	1.00000	.10915E+01	.66342E+00	.91515E-01
9	.98198	-.50901	1.00000	.10797E+01	.63025E+00	.79651E-01
10	.98399	-.50800	1.00000	.10701E+01	.59874E+00	.70092E-01
20	.99359	-.50321	1.00000	.10269E+01	.35849E+00	.26870E-01
30	.99680	-.50160	1.00000	.10132E+01	.21464E+00	.13234E-01
40	.99825	-.50087	1.00000	.10072E+01	.12851E+00	.71610E-01
50	.99901	-.50050	1.00000	.10041E+01	.76945E-01	.40621E-02
75	.99974	-.50013	1.00000	.10011E+01	.21344E-01	.10736E-02
100	.99993	-.50004	1.00000	.10003E+01	.59205E-02	.29409E-03

The Power Method also works for the case when the dominant eigenvalue is repeated but A still has a complete set of linearly independent eigenvectors. In this case, \vec{x}^k approaches a linear combination of the eigenvectors corresponding to the dominant eigenvalue.

Exercise Verify that the Power Method converges to a linear combination of the eigenvectors corresponding to the dominant eigenvalue when the dominant eigenvalue is repeated but A is not defective.

When the dominant eigenvalue is not unique, e.g., $\pm\rho(A)$, then the results oscillate between the eigenvector corresponding to $+\rho$ and $-\rho$.

Example Let A be the matrix

$$A = \begin{pmatrix} 57 & 153 & 144 \\ -30 & -84 & -84 \\ 9 & 27 & 30 \end{pmatrix},$$

where $\lambda(A) = \{6, -6, 3\}$, i.e., A does not have a unique dominant eigenvalue. As expected, in this case the computed iterates show no tendency towards converging rather they oscillate.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	1.00000	1.00000	1.00000
1	1.00000	-.55932	.18644
2	1.00000	-.76470	.29412
3	1.00000	-.54000	.16000
4	1.00000	-.74418	.30232
5	1.00000	-.53403	.15183
6	1.00000	-.74033	.30387
7	1.00000	-.53245	.14967
8	1.00000	-.73943	.30423
9	1.00000	-.53205	.14912
10	1.00000	-.73921	.30432

Inverse Power Method

Suppose we want to determine the smallest eigenvalue in magnitude. We know that if λ_i is an eigenvalue of A then $1/\lambda_i$ is an eigenvalue of A^{-1} ; thus one over the smallest eigenvalue of A in magnitude is the largest eigenvalue of A^{-1} .

Consequently we can simply use the Power Method with the matrix A^{-1} to determine the eigenvector corresponding to the smallest eigenvalue of A .

Inverse Power Method Let A have real eigenvalues and a complete set of linearly independent eigenvectors. Assume that the smallest eigenvalue in magnitude is unique (not repeated) and the eigenvalues are ordered as follows

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n| \implies \frac{1}{|\lambda_n|} > \frac{1}{|\lambda_{n-1}|} \geq \frac{1}{|\lambda_{n-2}|} \geq \cdots \geq \frac{1}{|\lambda_1|}$$

and the associated eigenvectors are denoted \vec{v}_i , $i = 1, \dots, n$.

Given \vec{x}^0 then we generate a sequence of iterates \vec{x}^k from

$$\vec{x}^k = \delta_k A^{-1} \vec{x}^{k-1}$$

where δ_k is a suitably chosen scaling factor.

How do we implement this method? We don't want to compute A^{-1} so instead we turn it into the problem of solving a linear system; we have

$$Ax^k = \delta_k x^{k-1}$$

To implement this algorithm we determine $A = LU$ or if A is symmetric either LL^T (A must be positive definite) or LDL^T and for each iteration we perform a back and forward solve. Convergence is analogous to the Power Method. Specifically we form $A = LU$ before the iteration starts then for each iteration we must do the following.

- solve $L\vec{y} = \vec{x}^{k-1}$ using a forward solve
- solve $U\vec{z} = \vec{y}$ using a back solve
- set $\vec{x}^k = \vec{z}/\delta_k$ for a scaling δ_k

The original factorization requires $\mathcal{O}(n^3)$ operations but each iteration is $\mathcal{O}(n^2)$.

Shifted Inverse Power Method

Sometimes we might want to find the eigenvalue nearest some number, say μ . In addition, we might want to determine a way to speed up slow convergence of the Power or Inverse Power Methods.

First note that if $A\vec{x} = \lambda\vec{x}$ then for the shifted matrix $A - \mu I$ we have

$$A\vec{x} = \lambda\vec{x} \implies A\vec{x} - \mu I\vec{x} = \lambda\vec{x} - \mu I\vec{x} \implies (A - \mu I)\vec{x} = (\lambda - \mu)\vec{x}$$

so the eigenvalues of $A - \mu I$ are just $\lambda - \mu$ and the eigenvectors are the same as A .

Suppose now that we want to find the eigenvalue of A that is nearest to μ . This means that we are looking for the *smallest* eigenvalue of $A - \mu I$ so we just apply the Inverse Power Method using the matrix $A - \mu I$.

When we use the Rayleigh Quotient we are finding an approximation to an eigenvalue of $(A - \mu I)^{-1}$ so we first take the reciprocal to get an approximation of an eigenvalue of $A - \mu I$ and then shift the result by μ because if σ is an eigenvalue

of $A - \mu I$ then $\sigma = \lambda - \mu \implies \lambda = \sigma + \mu$.

Rayleigh Quotient Iteration

The last variant of the Power Method which we describe converges very rapidly (cubically for symmetric matrices) but requires more work than the Inverse Power Method. In addition, it is in general impossible to predict to which eigenvector the iteration will converge. The basic idea is that we use the Shifted Inverse Power Method but we keep updating the shift. We can think of the initial value of μ as a guess for our eigenvalue.

When we get $\vec{x}^1 = \delta_1(A - \mu_0 I)^{-1} \vec{x}^0$ we can obtain an approximation to the eigenvalue of A and call it μ_1 . Then to get \vec{x}^k we use

$$\vec{x}^k = \delta_k(A - \mu_{k-1} I)^{-1} \vec{x}^{k-1}$$

Example The matrix

$$A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{pmatrix}$$

has eigenvalues $\lambda_1 = 6$, $\lambda_2 = 3$, and $\lambda_3 = 1$ and corresponding orthonormal eigenvectors

$$\vec{v}_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \quad \vec{v}_2 = \frac{1}{\sqrt{6}} \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix} \quad \vec{v}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} .$$

We apply the inverse power method having a fixed shift and the Rayleigh quotient iteration for which the shift is updated. For two different initial vectors, we give results for three choices of the inverse power method shift; the initial shift for the Rayleigh quotient iteration was chosen to be the same as that for the inverse power method.

Inverse Power Method					Rayleigh Quotient Iteration				
μ	k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	k	μ_k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
-1.0	5	.15453	-.60969	-.77743	2	3.00023	.82858	.37833	-.41269
-1.0	21	.00000	-.70711	-.70711	3	3.00000	.81659	.40825	-.40825
3.5	3	-.81945	-.40438	.40616	1	3.04678	-.88302	-.30906	.35321
3.5	7	-.81650	-.40825	.40825	3	3.00000	-.81650	-.40825	.40825
8.0	7	.56427	-.58344	.58411	3	5.99931	.56494	-.58309	.58382
8.0	17	.57735	-.57735	.57735	4	6.00000	.57735	-.57735	.57735
-1.0	5	.00053	.70657	.70765	2	1.00004	.00158	.70869	.70552
-1.0	10	.00000	.70711	.70711	3	1.00000	.00000	.70711	.70711
3.5	15	.25400	-.88900	-.38100	2	1.06818	.06742	.63485	.76969
3.5	30	-.81647	-.40821	.40834	4	1.00000	.00000	.70711	.70711
8.0	5	-.57735	.57459	-.58009	3	5.99970	-.57733	.57181	-.58285
8.0	11	-.57735	.57735	-.57735	4	6.00000	-.57735	.57735	-.57735

The first six rows of the table are for the initial vector $\frac{1}{\sqrt{29}}(2, 3, -4)^T$ and the three shifts $\mu = -1, 3.5,$ and 8 . We give results for certain iteration numbers k . The last six rows are analogous results for the initial vector $\frac{1}{\sqrt{74}}(7, 3, 4)^T$.

Notice that the Rayleigh Quotient Iteration is sensitive to the choice of the initial vector, not the shift.

Exercise How does the amount of work for each iteration for the Rayleigh Quotient Method differ from the work for the Inverse Power Method?

What if we want, e.g., the two largest eigenvalues of A ? Can we use the Power Method to obtain these?

There is a way to “deflate” your $n \times n$ matrix A to convert it to an $(n-1) \times (n-1)$ matrix which has the same eigenvalues of A except for the one you just found. However, this deflation technique can only be used a couple of times before too many errors are introduced. One approach would be to use these approximations to the eigenvalues as a starting guess in the Inverse Power Method.

The QR Algorithm for Finding All Eigenvalues & Eigenvectors

The methods we have described allow us to calculate a few eigenvalue and corresponding eigenvectors of a matrix. If one desires all or most of the eigenvalues, then the QR method is preferred. The basic QR method is very simple to describe. Starting with $A^{(0)} = A$, the sequence of matrices $A^{(k)}$, $k = 1, 2, \dots$, is determined by

For $k = 0, 1, 2, \dots$, set

$$A^{(k)} = Q^{(k+1)} R^{(k+1)} \quad \text{and} \quad A^{(k+1)} = R^{(k+1)} Q^{(k+1)} .$$

Thus, one step of the QR method consists of performing a QR factorization of the current iterate $A^{(k)}$ and then forming the new iterate by multiplying the factors in reverse order. Remarkably, as the following example illustrates, often $A^{(k)}$ tends to an upper triangular matrix which is unitarily similar to the original matrix.

Example The following are some iterates of the QR method applied to

$$A = A^{(0)} = \begin{pmatrix} 3 & -1 & 2/3 & 1/4 & -1/5 & 1/3 \\ 4 & 6 & -4/3 & 2 & 4/5 & -1/3 \\ 6 & -3 & -3 & -3/4 & 9/5 & 1/2 \\ 4 & 8 & -4/3 & -1 & 8/5 & 4/3 \\ 5 & 5 & 5 & 5/2 & 3 & 5/2 \\ 12 & -3 & 2 & 3 & 18/5 & 5 \end{pmatrix}$$

$$A^{(10)} = \begin{pmatrix} 9.086 & -.2100 & -2.101 & 7.536 & -.9124 & -10.06 \\ .7445 & 9.338 & 2.686 & -2.775 & -1.029 & -5.386 \\ .0955 & .0611 & -5.711 & .3987 & -5.218 & -6.456 \\ .0002 & .0004 & -.0024 & -3.402 & -.5699 & -1.777 \\ -4 & -4 & .0051 & .0850 & 2.885 & 3.257 \\ -9 & -10 & -8 & -6 & -6 & .8045 \end{pmatrix}$$

$$A^{(30)} = \begin{pmatrix} 9.694 & .3468 & -3.383 & 6.294 & .4840 & -.6447 \\ .1614 & 8.727 & -.5014 & 5.005 & -1.290 & -11.44 \\ -5 & -4 & -5.705 & .4308 & -5.207 & -6.370 \\ * & -11 & -7 & -3.395 & -.6480 & -1.814 \\ * & * & -8 & .0031 & 2.875 & 3.230 \\ * & * & * & * & * & .8045 \end{pmatrix}$$

$$A^{(60)} = \begin{pmatrix} 9.637 & .4494 & -3.254 & 5.380 & .6933 & 1.255 \\ .0088 & 8.784 & -1.054 & 5.977 & -1.190 & -11.39 \\ * & * & -5.705 & .4333 & -5.207 & -6.370 \\ * & * & * & -3.395 & -.6511 & -1.815 \\ * & * & * & -4 & 2.874 & 3.229 \\ * & * & * & * & * & .8045 \end{pmatrix}$$

The entry -4 , for example, indicates that that entry is less than 10^{-4} in magnitude; the entry $*$ indicates that that entry is less than 10^{-12} in magnitude. It takes over 90 iterations for all the diagonal entries to approximate eigenvalues to four significant figures; it takes over 300 iterations for all the subdiagonal entries to become less than 10^{-12} in magnitude.

The QR method as defined above is **impractical** for two reasons. First, each step of the method requires a QR factorization costs $O(n^3)$ multiplications and a like number of additions or subtractions. Second, we have only linear convergence of the subdiagonal entries of $A^{(k+1)}$ to zero. Thus, the method described here requires too many steps and each step is too costly. Fortunately, the approach has been modified to make it more computationally feasible.

The three essential ingredients in making the QR method practical are:

- the use of a preliminary reduction to upper Hessenberg form in order to reduce the cost per iteration;
- the use of a deflation procedure whenever a subdiagonal entry effectively vanishes, again in order to reduce the cost per iteration;
- and the use of a shift strategy in order to accelerate convergence.

We do not have time to go into the QR algorithm or its variants in detail but if you are interested, Dr. Gallivan in the Mathematics Department gives a course in Numerical Linear Algebra.

If by now you are thinking that you will never need eigenvalues, below is a discussion of the PageRank algorithm which is a patented Google algorithm for ranking pages. Note the next to last sentence where it describes what they are really doing is finding an eigenvector. Eigenvalues and eigenvectors occur in applications that you might not realize!

3.2 PageRank algorithm

Given a set of n web pages and the adjacency matrix A (defined previously), PageRank [Brin and Page, 1998] first constructs a probability transition matrix M by renormalizing each row of A to sum to 1. One then imagines a random web surfer who at each time step is at some web page, and decides which page to visit on the next step as follows: with probability $1 - \epsilon$, she randomly picks one of the hyperlinks on the current page, and jumps to the page it links to; with probability ϵ , she “resets” by jumping to a web page picked uniformly and at random from the collection.³ Here, ϵ is a parameter, typically set to 0.1-0.2. This process defines a Markov chain on the web pages, with transition matrix $\epsilon U + (1 - \epsilon)M$, where U is the transition matrix of uniform transition probabilities ($U_{ij} = 1/n$ for all i, j). The vector of PageRank scores p is then defined to be the stationary distribution of this Markov chain. Equivalently, p is the principal eigenvector of the transition matrix $(\epsilon U + (1 - \epsilon)M)^T$ (see, e.g. Golub and Van Loan, 1996), since by definition the stationary distribution satisfies

$$(\epsilon U + (1 - \epsilon)M)^T p = p. \quad (1)$$

The asymptotic chance of visiting page i , that is, p_i , is then taken to be the “quality” or authoritativeness of page i .