Lecture 15: Eigenvectors / Eigenvalues - Iterative Methods

June 25, 2025

Outline

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

- With the power iteration we can only recover q_1.
- What about other eigenvectors? We can find another eigenvector using the same idea of repeatedly multiplying a starting vector by a matrix.
- The inverse iteration can recover the eigenvector q_n associated with the **smallest** magnitude eigenvalue.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

- We are assuming throughout this lecture that A is invertible.
- This is OK, because our assumption from last time, that A is SPD, carries on throughout this lecture.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

- The inverse iteration instead multiplies the starting vector $v^{(0)}$ by A^{-1}.
- Note that if

$$\begin{aligned} Ax &= \lambda x, \text{ then} \\ x &= \lambda A^{-1} x, \text{ and so} \\ \frac{1}{\lambda} x &= A^{-1} x, \text{ assuming } \lambda \neq 0. \end{aligned}$$

- Therefore, the eigenvalues of A^{-1} are the reciprocals of the eigenvalues of A:

$$\text{If } \Lambda(A) = \{\lambda_i\}, \text{ then } \Lambda(A^{-1}) = \left\{ \frac{1}{\lambda_i} \right\}.$$

- We can therefore use this to find q_n instead of q_1.
- The proof follows the same steps as the one for the power iteration.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

Proof. We have that

$$
\begin{aligned}
A^{-1}v^{(0)} &= \frac{c_1 q_1}{\lambda_1} + \frac{c_2 q_2}{\lambda_2} + \ldots + \frac{c_n q_n}{\lambda_n} \\
A^{-k}v^{(0)} &= \frac{c_1 q_1}{(\lambda_1)^k} + \frac{c_2 q_2}{(\lambda_2)^k} + \ldots + \frac{c_n q_n}{(\lambda_n)^k} \\
&= \frac{1}{(\lambda_n)^k}\left(c_1 q_1 \left(\frac{\lambda_n}{\lambda_1}\right)^k + c_2 q_2 \left(\frac{\lambda_n}{\lambda_2}\right)^k + \ldots + c_n q_n\right)
\end{aligned}
$$

For large k,

$$
A^{-k}v^{(0)} \approx c_n \left(\frac{1}{\lambda_n}\right)^k q_n.
$$

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

- Since the eigenvectors are orthonormal, the scale factor doesn't matter.
- We can normalize to find q_n as

$$q_n \to \frac{A^{-k} v^{(0)}}{\|A^{-k} v^{(0)}\|} \quad \text{as } k \to \infty.$$

- This is the eigenvector for the the smallest magnitude eigenvalue.

$\square$

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration

- Note we don't actually form A^{-1}; we instead solve a linear system.
- The inverse iteration pseudocode is given in Algorithm 1.

Algorithm 1 (Basic) Inverse Iteration Algorithm

$v^{(0)} =$ initial guess, s.t. $\left\| v^{(0)} \right\| = 1$
for $k = 1, 2, \ldots$
  $w = A^{-1} v^{(k-1)}$ ▷ Actually, solve a linear system:
  ▷ $Aw = v^{(k-1)}$
  $v^{(k)} = \frac{w}{\|w\|}$
  $\lambda^{(k)} = \left(v^{(k)}\right)^T A v^{(k)}$ ▷ Rayleigh Quotient
end for

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

- The inverse iteration still only allows us compute one eigenvector, q_n.
- “Shifting” the eigenvalues will let us find more.
- The idea is to use the fact that the smallest possible magnitude eigenvalue is zero!
- We then try to modify A so the “target” eigenvector has the smallest magnitude eigenvalue near zero.
- Therefore, the inverse iteration will find this target eigenvector.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

- Consider $B = A - \mu I$, with $\mu \neq 0$ not an eigenvalue.
- If A's eigenvalues/vectors are known, what are B's?
- Since

$$\begin{aligned} Ax &= \lambda x, \text{ we have} \\ Ax - \mu x &= \lambda x - \mu x, \text{ and so} \\ \underbrace{(A - \mu I)}_{B} x &= (\lambda - \mu) x. \end{aligned}$$

- Therefore, for the matrix B we have that the
  1. eigenvectors are the same,
  2. eigenvalues are shifted: $\lambda_j - \mu$ for $\lambda_j \in \Lambda(A)$.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

- If we (somehow) expect λ_j close to μ, then $\lambda_j - \mu$ is the smallest magnitude eigenvalue of B.
- We can then apply the inverse iteration to find its eigenvector q_j.
- This is an advantage of the inverse iteration over the power iteration.
- With the inverse iteration we can select the specific eigenvector to recover, **if** we can choose μ close to the corresponding λ_j.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

- The inverse iteration has linear convergence behaviour, as stated in the next theorem.

Theorem 1 (Inverse iteration, with shifting, convergence)

Suppose λ_J is the closest eigenvalue to μ and λ_L is the next closest, or $|\mu - \lambda_J| < |\mu - \lambda_L| \leq |\mu - \lambda_j|$, for $j \neq J$, and $q_J^T v^{(0)} \neq 0$.
Then

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

as $k \to \infty$.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

- So convergence depends on ratios of **shifted** eigenvalues rather than original eigenvalues.
- If we unluckily choose $v^{(0)}$ orthogonal to q_J, so that $q_J^T v^{(0)} = 0$, we will not get convergence; instead we will get to max-iterations, without any answer. In this case, we can make a different guess for $v^{(0)}$, and try again.

Eigenvectors / Eigenvalues - Iterative Methods - Inverse iteration - Shifting Eigenvalues

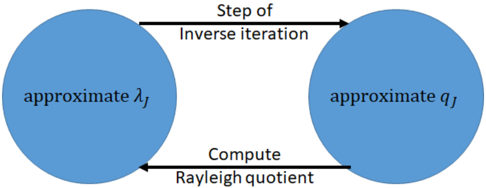
Algorithm 2 gives pseudocode for the shifted inverse iteration.

Algorithm 2 (Shifted) Inverse Iteration Algorithm

For given μ:
$v^{(0)} =$ initial guess, s.t. $\left\| v^{(0)} \right\| = 1$
for $k = 1, 2, \ldots$
  Solve $(A - \mu I)w = v^{(k-1)}$
  $v^{(k)} = \frac{w}{\|w\|}$
  $\lambda^{(k)} = \left(v^{(k)}\right)^T A v^{(k)}$ ▷ Rayleigh Quotient
end for

Eigenvectors / Eigenvalues - Iterative Methods - Rayleigh Quotient iteration

- The shifted inverse iteration needs an estimate of the eigenvalue λ_j to use for μ.
- Observe the following:
  1. **Rayleigh quotient**, $r(v)$, estimates an eigenvalue given its approximate eigenvector v (with quadratic convergence).
  2. **Inverse iteration** estimates an eigenvector given the approximate eigenvalue μ (with linear convergence, by Theorem [Inverse Iteration Convergence]).
- The Rayleigh quotient iteration combines the two!
- It is the inverse iteration that updates μ with the latest guess for λ_j at each step (see figure below).



Eigenvectors / Eigenvalues - Iterative Methods - Rayleigh Quotient iteration

Pseudocode for the Rayleigh quotient iteration is given in Algorithm 3.

Algorithm 3 Rayleigh Quotient Iteration Algorithm

$v^{(0)} =$ initial guess, s.t. $\left\| v^{(0)} \right\| = 1$

$\lambda^{(0)} = \left(v^{(0)}\right)^T A v^{(0)} \; (= r(v^{(0)}))$ ▷ Rayleigh Quotient

for $k = 1, 2, \ldots$

  Solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$ ▷ Using current λ estimate
  ▷ instead of fixed initial μ

  $v^{(k)} = \frac{w}{\|w\|}$

  $\lambda^{(k)} = \left(v^{(k)}\right)^T A v^{(k)}$ ▷ Rayleigh Quotient

end for

Eigenvectors / Eigenvalues - Iterative Methods - Rayleigh Quotient iteration

- The convergence when combining the Rayliegh quotient and the inverse iteration (i.e., Rayliegh quotient iteration) is cubic.

Theorem 2 (Rayleigh quotient iteration convergence)

RQI converges cubically for "almost all" starting vectors $v^{(0)}$. That is,

$$\left\| v^{(k+1)} - (\pm q_J) \right\| = O\left(\left\| v^{(k)} - (\pm q_J) \right\|^3 \right),$$

and

$$|\lambda^{(k+1)} - \lambda_J| = O\left(|\lambda^{(k)} - \lambda_J|^3 \right).$$

- In practice, each iteration roughly triples the number of digits of accuracy.

Remarks:

1. Note that "almost all" includes at least $q_J^T v^{(0)} \neq 0$.

Eigenvectors / Eigenvalues - Iterative Methods - Rayleigh Quotient iteration

- For example, consider this matrix:

$$A = \begin{bmatrix} 21 & 7 & -1 \\ 5 & 7 & 7 \\ 4 & -4 & 20 \end{bmatrix}, \quad v^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

- The estimated eigenvalue with the Rayleigh quotient iteration is ($\lambda^{(0)}$ as initial guess)

$$\begin{aligned}
\lambda^{(0)} &= 22 & v^{(0)} &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T, \\
\lambda^{(1)} &= 24.0802 & v^{(1)} &= \begin{bmatrix} 0.8655 & 0.3619 & 0.3462 \end{bmatrix}^T, \\
\lambda^{(2)} &= 24.0013 & v^{(2)} &= \begin{bmatrix} -0.8169 & -0.4079 & -0.4077 \end{bmatrix}^T, \\
\lambda^{(3)} &= 24.00000017 & v^{(3)} &= \begin{bmatrix} 0.8164 & 0.4082 & 0.4082 \end{bmatrix}^T.
\end{aligned}$$

Remarks:

1. It is weird, but correct (verify it for yourself), that we get the negative of the previous and future eigenvectors, as $v^{(2)}$.

Eigenvectors / Eigenvalues - Iterative Methods - Computational Complexity

The following gives the operation counts for each of the iterative methods discussed so far.

- **Power iteration**: each step involved $Av^{(k-1)} \rightarrow O(n^2)$ flops.
- **Inverse iteration**: each step requires solving $(A - \mu I)w = v^{(k-1)}$. This **would** be $O(n^3)$ per step, however, we can pre-factor into L and U at the start (Recall, at cost: $\frac{2}{3}n^3 + O(n^2)$). Then we just do forward/backward solves for each iteration. Hence, we have $O(n^2)$ flops per step.
- **Rayleigh quotient iteration**: Matrix $A - \lambda^{(k-1)}I$ changes at each step so we can not pre-factor. Therefore, for RQI we have $O(n^3)$ flops per iteration.
- For all three of the methods, if A is tridiagonal the operation counts reduce to $O(n)$ flops.

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

- In the previous sections we looked at the power, inverse, and Rayleigh quotient iterations for finding a **single** eigenvector/eigenvalue.
- Our next goal is to find **more than one** eigenvector/eigenvalue pair at a time.
- First some definitions are needed.

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

Definition 4.1
Matrices A and B are **similar** *if $B = X^{-1}AX$ for some non-singular X.*

Definition 4.2
If $X \in \mathbb{R}^{n\times n}$ is nonsingular, then $A \to X^{-1}AX$ is called a **similarity transformation** *of A.*

Theorem 3
If matrices A and B are similar, then they have the same characteristic polynomial, and hence the same eigenvalues.

Proof.
See Lecture Notes. $\square$

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

- The idea is to apply a sequence of similarity transformations to A that converge to a **diagonal** matrix, which has the eigenvalues on its diagonal.
- Recall that we are only considering real symmetric matrices.
- To achieve this we will rely on the QR factorization again.
- **Fun Fact:** In 2000, the QR Iteration was named one of the top ten most important algorithms for science and engineering developed in the 20th century.

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

- Given $A^{(k-1)}$, we factor it:

$$\begin{aligned} A^{(k-1)} &= Q^{(k)} R^{(k)}, \text{ so that} \\ \left(Q^{(k)}\right)^T A^{(k-1)} &= R^{(k)}. \end{aligned}$$

- Defining the next matrix, $A^{(k)}$, as $R^{(k)} Q^{(k)}$, then yields a similarity transformation:

$$\begin{aligned} A^{(k)} &:= R^{(k)} Q^{(k)} \\ &= \left(Q^{(k)}\right)^T A^{(k-1)} Q^{(k)}. \end{aligned}$$

- Therefore, $A^{(k-1)}$ and $A^{(k)}$ are similar.

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

The QR Iteration simply repeats this process as shown in Algorithm 4.

Algorithm 4 Basic QR Iteration

$A^{(0)} = A$
for $k = 1, 2, \ldots$
  $Q^{(k)} R^{(k)} = A^{(k-1)}$ ▷ Compute QR factors of $A^{(k-1)}$
  $A^{(k)} = R^{(k)} Q^{(k)}$ ▷ "Recombine" in reverse order
end for

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

- That's it!
- We just compute the QR factorization of $A^{(k-1)} = QR$ and reverse the order to construct $A^{(k)} = RQ$.
- Eventually $A^{(k)}$ becomes diagonal, with the eigenvalues of A on the diagonal.
- As $A^{(k)}$ converges to eigenvalues on the diagonal (we will justify why this happens in the next lecture), the product of the $Q^{(k)}$'s gives the set of eigenvectors.
- That is, denoting

$$\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \dots Q^{(k)},$$

we have the relation

$$A^{(k)} = \left(\underline{Q}^{(k)}\right)^T A \underline{Q}^{(k)}.$$

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

Consider the following QR Iteration example with

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix} = A^{(0)}.$$

One can verify (say using Matlab) that A is SPD.
We can use this Matlab code to compute the first three QR iterations:

```
A0 = [2 1 1 ; 1 3 1 ; 1 1 4];
[Q1,R1] = qr(A0);
A1 = R1 * Q1;
[Q2,R2] = qr(A1);
A2 = R2 * Q2;
[Q3,R3] = qr(A2);
A3 = R3 * Q3;
```

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

Our results are then

$$A^{(1)} = \begin{bmatrix} 4.1667 & 1.0954 & -1.2671 \\ 1.0954 & 2.0000 & 0.0000 \\ -1.2671 & 0.0000 & 2.8333 \end{bmatrix},$$

$$A^{(2)} = \begin{bmatrix} 5.0909 & 0.1574 & 0.6232 \\ 0.1574 & 1.8618 & -0.5470 \\ 0.6232 & -0.5470 & 2.0473 \end{bmatrix},$$

$$A^{(3)} = \begin{bmatrix} 5.1987 & -0.0759 & -0.2073 \\ -0.0759 & 2.1818 & 0.4966 \\ -0.2073 & 0.4966 & 1.6195 \end{bmatrix}.$$

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

- The true solution for this matrix is $\Lambda(A) = \{5.2143, 2.4608, 1.3249\}$.
- At each iteration above the off-diagonals get closer to zero.
- Moreover, the diagonal entires are converging to the true eigenvalues.

Eigenvectors / Eigenvalues - Iterative Methods - QR Iteration

Remarks:

1. QR Iteration is mathematically sound, but not good computationally.
2. Do not implement this algorithm!
3. In the next lecture we will discuss an equivalent algorithm that is computationally better.