Algorithm Foundations of Data Science and Engineering

Lecture 4: Computation of Eigenvalue and Eigenvector

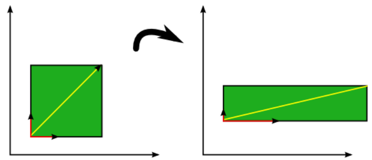
MING GAO

DaSE @ ECNU
(for course related communications)
mgao@dase.ecnu.edu.cn

Mar. 18, 2019

Outline

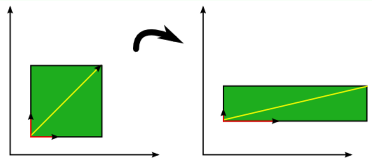
Transformation



Given a matrix A

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

Transformation

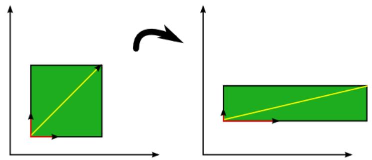


Given a matrix A

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

- The transformation in this case is a simple scaling with factor 2 in the horizontal direction and factor 0.5 in the vertical direction.

Transformation

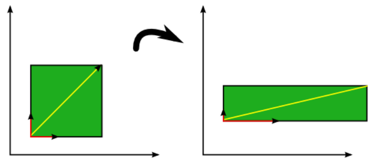


Given a matrix A

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

- The transformation in this case is a simple scaling with factor 2 in the horizontal direction and factor 0.5 in the vertical direction.
- A vector $\mathbf{v} = (x, y)$ is then scaled by applying this transformation as $\mathbf{v}' = A\mathbf{v}$.

Transformation

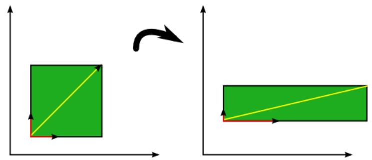


Given a matrix A

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

- The transformation in this case is a simple scaling with factor 2 in the horizontal direction and factor 0.5 in the vertical direction.
- A vector $\mathbf{v} = (x, y)$ is then scaled by applying this transformation as $\mathbf{v}' = A\mathbf{v}$.
- The above figure shows that the direction of some vectors (shown in red) is not affected by this linear transformation.

Transformation



Given a matrix A

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

- The transformation in this case is a simple scaling with factor 2 in the horizontal direction and factor 0.5 in the vertical direction.
- A vector $\mathbf{v} = (x, y)$ is then scaled by applying this transformation as $\mathbf{v}' = A\mathbf{v}$.
- The above figure shows that the direction of some vectors (shown in red) is not affected by this linear transformation.
- Eigenvectors (red) do not change direction when a linear transformation is applied to them. Other vectors (yellow) do.

Eigenvalue and eigenvector

Definition

Given a matrix $A \in \mathbb{R}^{n\times n}$, and non-zero column vector $\mathbf{v}$, if

$$A\mathbf{v} = \lambda\mathbf{v},$$

where λ is an eigenvalue of A and $\mathbf{v}$ is an eigenvector corresponding to eigenvalue λ.

Eigenvalue and eigenvector

Definition

Given a matrix $A \in \mathbb{R}^{n \times n}$, and non-zero column vector $\mathbf{v}$, if

$$A\mathbf{v} = \lambda\mathbf{v},$$

where λ is an eigenvalue of A and $\mathbf{v}$ is an eigenvector corresponding to eigenvalue λ.

- Transformation: a matrix A acts on vectors $\mathbf{v}$ like a function does, with input $\mathbf{v}$ and output $A\mathbf{v}$. Eigenvectors are vectors for which $A\mathbf{v}$ is parallel to $\mathbf{v}$. In other words: $A\mathbf{v} = \lambda\mathbf{v}$.

Eigenvalue and eigenvector

Definition

Given a matrix $A \in \mathbb{R}^{n\times n}$, and non-zero column vector $\mathbf{v}$, if

$$A\mathbf{v} = \lambda\mathbf{v},$$

where λ is an eigenvalue of A and $\mathbf{v}$ is an eigenvector corresponding to eigenvalue λ.

- Transformation: a matrix A acts on vectors $\mathbf{v}$ like a function does, with input $\mathbf{v}$ and output $A\mathbf{v}$. Eigenvectors are vectors for which $A\mathbf{v}$ is parallel to $\mathbf{v}$. In other words: $A\mathbf{v} = \lambda\mathbf{v}$.
- $A\mathbf{v} = \lambda\mathbf{v}$ can be rewrote as

$$A\mathbf{v} - \lambda\mathbf{v} = \mathbf{0}, \text{ i.e., } (A - \lambda I)\mathbf{v} = \mathbf{0},$$

where I is the identity matrix of the same dimensions as A.

Applications

Many important applications in computer vision and machine learning, e.g.

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)
- PCA (Principal Component Analysis) for dimensionality reduction

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)
- PCA (Principal Component Analysis) for dimensionality reduction
- EigenFaces for face recognition

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)
- PCA (Principal Component Analysis) for dimensionality reduction
- EigenFaces for face recognition
- Graph robustness: algebraic connectivity

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)
- PCA (Principal Component Analysis) for dimensionality reduction
- EigenFaces for face recognition
- Graph robustness: algebraic connectivity
- Eigendecomposition forms the base of the geometric interpretation of covariance matrices

Applications

Many important applications in computer vision and machine learning, e.g.

- Singular value decomposition (SVD)
- PCA (Principal Component Analysis) for dimensionality reduction
- EigenFaces for face recognition
- Graph robustness: algebraic connectivity
- Eigendecomposition forms the base of the geometric interpretation of covariance matrices

How can we use computers to find eigenvalues and eigenvectors efficiently?

Directed approach

Given a matrix A

$$A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$$

For a non-zero column vector $\mathbf{v}$, equation $(A - \lambda I)\mathbf{v} = \mathbf{0}$ can only be defined if matrix $A - \lambda I$ is not invertible.

Directed approach

Given a matrix A

$$A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$$

For a non-zero column vector $\mathbf{v}$, equation $(A - \lambda I)\mathbf{v} = \mathbf{0}$ can only be defined if matrix $A - \lambda I$ is not invertible.

- If a square matrix is not invertible, that means that its determinant must equal zero.

Directed approach

Given a matrix A

$$A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$$

For a non-zero column vector $\mathbf{v}$, equation $(A - \lambda I)\mathbf{v} = \mathbf{0}$ can only be defined if matrix $A - \lambda I$ is not invertible.

- If a square matrix is not invertible, that means that its determinant must equal zero.
- Therefore, to find the eigenvectors of A, we simply have to solve the following equation (characteristic equation):

$$Det(A - \lambda I) = 0.$$

Directed approach

Given a matrix A

$$A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$$

For a non-zero column vector $\mathbf{v}$, equation $(A - \lambda I)\mathbf{v} = \mathbf{0}$ can only be defined if matrix $A - \lambda I$ is not invertible.

- If a square matrix is not invertible, that means that its determinant must equal zero.
- Therefore, to find the eigenvectors of A, we simply have to solve the following equation (characteristic equation):

$$Det(A - \lambda I) = 0.$$

- For above example, we have

$$Det(A - \lambda I) = \lambda^2 - 3\lambda + 2 = 0.$$

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A - \lambda_i I)\mathbf{v} = \mathbf{0}$

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A - \lambda_i I)\mathbf{v} = \mathbf{0}$
For the first eigenvalue $\lambda_1 = 1$, we have a system of equations

$$\left(\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \mathbf{0}.$$

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A - \lambda_i I)\mathbf{v} = \mathbf{0}$
For the first eigenvalue $\lambda_1 = 1$, we have a system of equations

$$\left(\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \mathbf{0}.$$

- That is $\begin{cases} 3x_{11} - 2x_{12} = x_{11} \\ x_{11} = x_{12} \end{cases}$

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A - \lambda_i I)\mathbf{v} = \mathbf{0}$
For the first eigenvalue $\lambda_1 = 1$, we have a system of equations

$$\left(\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \mathbf{0}.$$

- That is $\begin{cases} 3x_{11} - 2x_{12} = x_{11} \\ x_{11} = x_{12} \end{cases}$
- For this example, we arbitrarily choose $x_{11} = 1$. Thus, the eigenvector corresponds to eigenvalue $\lambda_1 = 1$ is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A-\lambda_i I)\mathbf{v} = \mathbf{0}$
For the first eigenvalue $\lambda_1 = 1$, we have a system of equations

$$\left(\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \mathbf{0}.$$

- That is $\begin{cases} 3x_{11} - 2x_{12} = x_{11} \\ x_{11} = x_{12} \end{cases}$
- For this example, we arbitrarily choose $x_{11} = 1$. Thus, the eigenvector corresponds to eigenvalue $\lambda_1 = 1$ is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
- Similarly, the eigenvector corresponds to eigenvalue $\lambda_2 = 2$ is $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$.

Calculating the eigenvectors

We can find eigenvector via solving the linear equation $(A-\lambda_i I)\mathbf{v} = \mathbf{0}$
For the first eigenvalue $\lambda_1 = 1$, we have a system of equations

$$\left(\begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} = \mathbf{0}.$$

- That is $\begin{cases} 3x_{11} - 2x_{12} = x_{11} \\ x_{11} = x_{12} \end{cases}$
- For this example, we arbitrarily choose $x_{11} = 1$. Thus, the eigenvector corresponds to eigenvalue $\lambda_1 = 1$ is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
- Similarly, the eigenvector corresponds to eigenvalue $\lambda_2 = 2$ is $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$.

However, this approach is not scalable.

Outline

Power method

Assume that for a matrix A there is a unique (ie only one) largest eigenvalue λ_1, i.e.,

$$\lambda_1 = \max_i \lambda_i$$

Then we can find λ_1 by the Power method.

Power method

Assume that for a matrix A there is a unique (ie only one) largest eigenvalue λ_1, i.e.,

$$\lambda_1 = \max_i \lambda_i$$

Then we can find λ_1 by the Power method.

- The power method is an iterative algorithm which has the following basic form for generating a single eigenvalue and eigenvector of A.

  1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
  2: For $k = 1, 2, \cdots$;
  3: Let $\mathbf{v} = A\mathbf{x}^{(k-1)}$;
  4: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;

Power method

Assume that for a matrix A there is a unique (ie only one) largest eigenvalue λ_1, i.e.,

$$\lambda_1 = \max_i \lambda_i$$

Then we can find λ_1 by the Power method.

- The power method is an iterative algorithm which has the following basic form for generating a single eigenvalue and eigenvector of A.

  1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
  2: For $k = 1, 2, \cdots$;
  3: Let $\mathbf{v} = A\mathbf{x}^{(k-1)}$;
  4: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;

  - It terminates when some convergence conditions are satisfied.

Power method

Assume that for a matrix A there is a unique (ie only one) largest eigenvalue λ_1, i.e.,

$$\lambda_1 = \max_i \lambda_i$$

Then we can find λ_1 by the Power method.

- The power method is an iterative algorithm which has the following basic form for generating a single eigenvalue and eigenvector of A.

  1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
  2: For $k = 1, 2, \cdots$;
  3: Let $\mathbf{v} = A\mathbf{x}^{(k-1)}$;
  4: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;

  - It terminates when some convergence conditions are satisfied.
- It can fail if there is not a single largest eigenvalue, i.e., $\lambda_1 = \lambda_2$.

Analysis of Power method

Why is it correct?

Analysis of Power method

Why is it correct?

- Recall that normalised eigenvectors form an orthonormal set, i.e., $\mathbf{v}_i^T \mathbf{v}_i = 1$, but $\mathbf{v}_i^T \mathbf{v}_j = 0$ for all $i \neq j$.

Analysis of Power method

Why is it correct?

- Recall that normalised eigenvectors form an orthonormal set, i.e., $\mathbf{v}_i^T \mathbf{v}_i = 1$, but $\mathbf{v}_i^T \mathbf{v}_j = 0$ for all $i \neq j$.
- Let $\mathbf{x}^{(0)}$ be an arbitrarily chosen vector in the subspace generated by the eigenvectors, i.e., $\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \mathbf{v}_i$ with $c_1 \neq 0$.

Analysis of Power method

Why is it correct?

- Recall that normalised eigenvectors form an orthonormal set, i.e., $\mathbf{v}_i^T \mathbf{v}_i = 1$, but $\mathbf{v}_i^T \mathbf{v}_j = 0$ for all $i \neq j$.
- Let $\mathbf{x}^{(0)}$ be an arbitrarily chosen vector in the subspace generated by the eigenvectors, i.e., $\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \mathbf{v}_i$ with $c_1 \neq 0$.Then $\mathbf{x}^{(k)}$ converges to the dominant eigenvector $\mathbf{v}_1$ because

Analysis of Power method

Why is it correct?

- Recall that normalised eigenvectors form an orthonormal set, i.e., $\mathbf{v}_i^T \mathbf{v}_i = 1$, but $\mathbf{v}_i^T \mathbf{v}_j = 0$ for all $i \neq j$.
- Let $\mathbf{x}^{(0)}$ be an arbitrarily chosen vector in the subspace generated by the eigenvectors, i.e., $\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \mathbf{v}_i$ with $c_1 \neq 0$.Then $\mathbf{x}^{(k)}$ converges to the dominant eigenvector $\mathbf{v}_1$ because

$$\lim_{k\to\infty} A^k \mathbf{x}^{(0)} = \lim_{k\to\infty} \sum_{i=1}^{n} c_i A^k \mathbf{v}_i = \lim_{k\to\infty} \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{v}_i$$

$$= \lim_{k\to\infty} c_1 \lambda_1^k \big[\mathbf{v}_1 + \sum_{i=2}^{n} \frac{c_i}{c_1} \big(\frac{\lambda_i}{\lambda_1}\big)^k \mathbf{v}_i\big] = c_1 \lambda_1^k \mathbf{v}_1$$

Analysis of Power method

Why is it correct?

- Recall that normalised eigenvectors form an orthonormal set, i.e., $\mathbf{v}_i^T \mathbf{v}_i = 1$, but $\mathbf{v}_i^T \mathbf{v}_j = 0$ for all $i \neq j$.
- Let $\mathbf{x}^{(0)}$ be an arbitrarily chosen vector in the subspace generated by the eigenvectors, i.e., $\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \mathbf{v}_i$ with $c_1 \neq 0$.Then $\mathbf{x}^{(k)}$ converges to the dominant eigenvector $\mathbf{v}_1$ because

$$
\begin{aligned}
\lim_{k\to\infty} A^k \mathbf{x}^{(0)} = \lim_{k\to\infty} \sum_{i=1}^{n} c_i A^k \mathbf{v}_i &= \lim_{k\to\infty} \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{v}_i \\
&= \lim_{k\to\infty} c_1 \lambda_1^k \Big[\mathbf{v}_1 + \sum_{i=2}^{n} \frac{c_i}{c_1} \Big(\frac{\lambda_i}{\lambda_1}\Big)^k \mathbf{v}_i \Big] = c_1 \lambda_1^k \mathbf{v}_1
\end{aligned}
$$

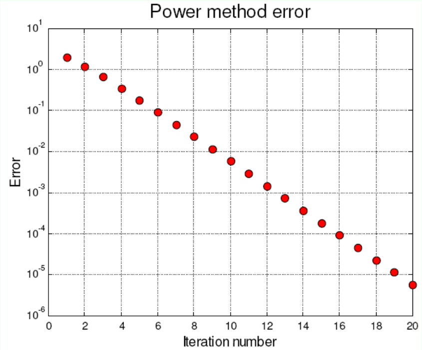
- It will find only one eigenvalue (the one with the greatest absolute value).

Analysis of Power method Cont'd

- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^k$.

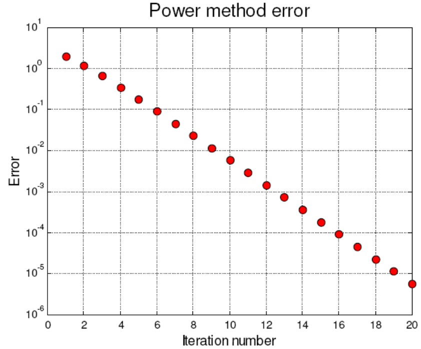
Analysis of Power method Cont'd

- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^k$.
- Since the reduction per iteration is $\left|\frac{\lambda_2}{\lambda_1}\right|$, that is, linear convergence.



Analysis of Power method Cont'd

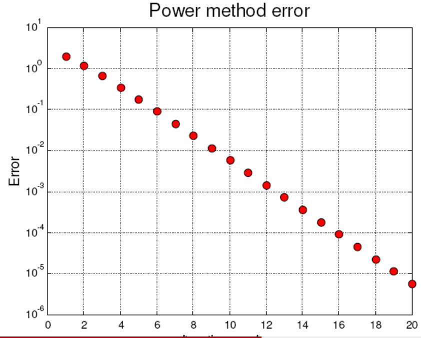
- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^k$.
- Since the reduction per iteration is $\left|\frac{\lambda_2}{\lambda_1}\right|$, that is, linear convergence.



- The Power method can be used to find the dominant eigenvalue of a symmetric matrix.

Analysis of Power method Cont'd

- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^k$.
- Since the reduction per iteration is $\left|\frac{\lambda_2}{\lambda_1}\right|$, that is, linear convergence.



- The Power method can be used to find the dominant eigenvalue of a symmetric matrix.
- The method does work if the dominant eigenvalue has multiplicity r. The estimated eigenvector will then be a linear combination of the r eigenvectors.

The power method Extension

- Inverse power method: it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.

The power method Extension

- Inverse power method: it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.
- Spectral shift: using the fact that the eigenvalues of $A - \alpha I$ are $\lambda_i - \alpha$. If we find the largest eigenvalue λ_1, we can find the largest in absolute value of $\lambda_i - \lambda_1$. However, it is not clear how it could be implemented in general to find all the eigenvalues of matrix A.

Inverse iteration

The inverse iteration method is a natural generalization of the power method.

Inverse iteration

The inverse iteration method is a natural generalization of the power method.

- For an invertible matrix A, it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.

Inverse iteration

The inverse iteration method is a natural generalization of the power method.

- For an invertible matrix A, it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.
- In general, though, the inverse matrix is not given, and calculating it is a computationally expensive operation.

Inverse iteration

The inverse iteration method is a natural generalization of the power method.

- For an invertible matrix A, it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.
- In general, though, the inverse matrix is not given, and calculating it is a computationally expensive operation. However, computing $x = A^{-1}b$ is equivalent to solving the system $Ax = b$ for x given b, and this operation can be efficiently performed.

Inverse iteration

The inverse iteration method is a natural generalization of the power method.

- For an invertible matrix A, it operates with A^{-1} rather than A since the eigenvalues of A^{-1} are $\frac{1}{\lambda_i}$. It gives a way of finding the smallest (in absolute value) eigenvalue of a matrix.
- In general, though, the inverse matrix is not given, and calculating it is a computationally expensive operation. However, computing $x = A^{-1}b$ is equivalent to solving the system $Ax = b$ for x given b, and this operation can be efficiently performed.

1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
2: For $k = 1, 2, \cdots$;
3:     Solve $A\mathbf{v} = \mathbf{x}^{(k-1)}$ for $\mathbf{v}$;
4:     Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;

Inverse iteration Cont'd

The advantage of inverse iteration is that it can be easily adapted to find any eigenvalue of the matrix A, instead of just the extreme ones.

Inverse iteration Cont'd

The advantage of inverse iteration is that it can be easily adapted to find any eigenvalue of the matrix A, instead of just the extreme ones.

- Observe that for any $\mu \in R$, the matrix $B = A - \mu I$ has eigenvalues $\{\lambda_1 - \mu, \cdots, \lambda_n - \mu\}$.

Inverse iteration Cont'd

The advantage of inverse iteration is that it can be easily adapted to find any eigenvalue of the matrix A, instead of just the extreme ones.

- Observe that for any $\mu \in R$, the matrix $B = A - \mu I$ has eigenvalues $\{\lambda_1 - \mu, \cdots, \lambda_n - \mu\}$.In particular, by choosing μ to be close to an eigenvalue λ_j of A, we can ensure that $\lambda_j - \mu$ is the eigenvalue of B of smallest magnitude.

Inverse iteration Cont'd

The advantage of inverse iteration is that it can be easily adapted to find any eigenvalue of the matrix A, instead of just the extreme ones.

- Observe that for any $\mu \in R$, the matrix $B = A - \mu I$ has eigenvalues $\{\lambda_1 - \mu, \cdots, \lambda_n - \mu\}$.In particular, by choosing μ to be close to an eigenvalue λ_j of A, we can ensure that $\lambda_j - \mu$ is the eigenvalue of B of smallest magnitude.

1: Pick some μ close the desired eigenvalue;
2: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
3: For $k = 1, 2, \cdots$;
4: Solve $(A - \mu I)\mathbf{v} = \mathbf{x}^{(k-1)}$ for $\mathbf{v}$;
5: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;

Outline

Newton iteration

Using linear Approximations to solve equations $f(x) = 0$, where $f(x)$ is a well-behaved function (differentiable). Let r be a root of the equation $f(x) = 0$.

Newton iteration

Using linear Approximations to solve equations $f(x) = 0$, where $f(x)$ is a well-behaved function (differentiable). Let r be a root of the equation $f(x) = 0$.

- Each iteration of the Newton iteration returns an approximate and improved root.

Newton iteration

Using linear Approximations to solve equations $f(x) = 0$, where $f(x)$ is a well-behaved function (differentiable). Let r be a root of the equation $f(x) = 0$.

- Each iteration of the Newton iteration returns an approximate and improved root.

  1: Pick a starting point x_0;

  2: Until the convergence condition;

  4: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$;

Newton iteration

Using linear Approximations to solve equations $f(x) = 0$, where $f(x)$ is a well-behaved function (differentiable). Let r be a root of the equation $f(x) = 0$.

- Each iteration of the Newton iteration returns an approximate and improved root.

  1: Pick a starting point x_0;
  2: Until the convergence condition;
  4: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$;

- Since the true root is r, and $h = r - x_0$, the number h measures how far the estimate x_0 is from the truth.

Newton iteration

Using linear Approximations to solve equations $f(x) = 0$, where $f(x)$ is a well-behaved function (differentiable). Let r be a root of the equation $f(x) = 0$.

- Each iteration of the Newton iteration returns an approximate and improved root.

  1: Pick a starting point x_0;

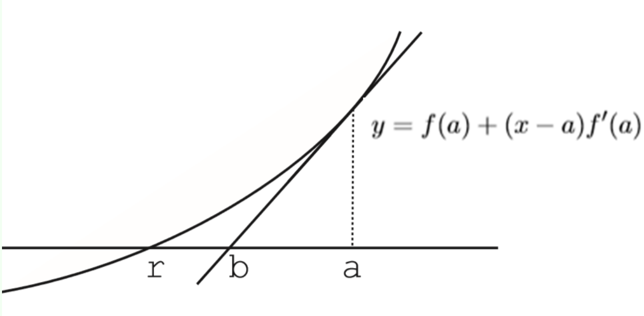
  2: Until the convergence condition;

  4: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$;

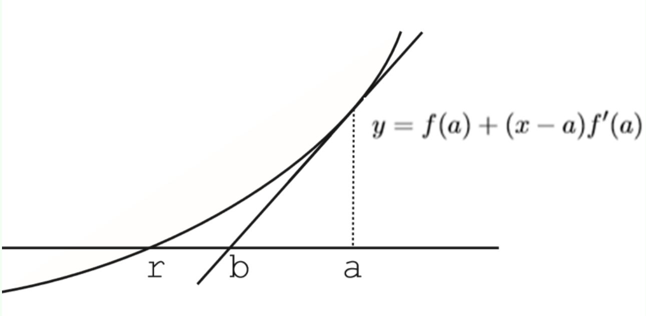
- Since the true root is r, and $h = r - x_0$, the number h measures how far the estimate x_0 is from the truth.
- Since h is 'small', we can use the linear (tangent line) approximation to conclude that

$$0 = f(r) = f(x_0+h) \approx f(x_0)+hf'(x_0) = f(x_0)+\frac{df(x)}{dx}|_{x=x_0}\triangle(x_0)$$

Newton iteration Cont'd

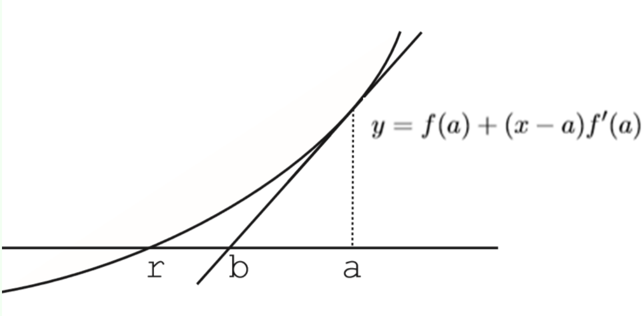


Newton iteration Cont'd



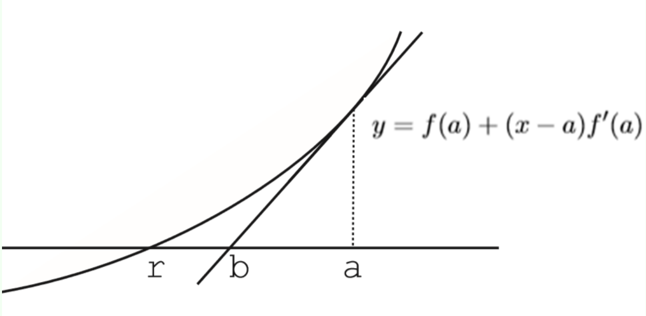
- Let a be the current estimate of r.

Newton iteration Cont'd



- Let a be the current estimate of r.
- Let b be the x-intercept of the tangent line. Then $b = a - \frac{f(a)}{f'(a)}$.

Newton iteration Cont'd



- Let a be the current estimate of r.
- Let b be the x-intercept of the tangent line. Then $b = a - \frac{f(a)}{f'(a)}$.
- b is just the next Newton estimate of r.

Rayleigh Quotient method

The inverse iteration method can be improved if we drop the restriction that the shift value remains constant throughout the iterations.

Rayleigh Quotient method

The inverse iteration method can be improved if we drop the restriction that the shift value remains constant throughout the iterations.

- Each iteration of the shifted inverse iteration method returns an approximate eigenvector, given an estimate of an eigenvalue.

Rayleigh Quotient method

The inverse iteration method can be improved if we drop the restriction that the shift value remains constant throughout the iterations.

- Each iteration of the shifted inverse iteration method returns an approximate eigenvector, given an estimate of an eigenvalue.

1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
2: Let $\lambda^{(0)} = r(\mathbf{x}^{(0)}) = (\mathbf{x}^{(0)})^T A\mathbf{x}^{(0)}$;
3: For $k = 1, 2, \cdots$;
4: Solve $(A - \lambda^{(k-1)} I)\mathbf{v} = \mathbf{x}^{(k-1)}$ for $\mathbf{v}$;
5: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;
6: Let $\lambda^{(k)} = r(\mathbf{x}^{(k)}) = (\mathbf{x}^{(k)})^T A\mathbf{x}^{(k)}$;

Rayleigh Quotient method

The inverse iteration method can be improved if we drop the restriction that the shift value remains constant throughout the iterations.

- Each iteration of the shifted inverse iteration method returns an approximate eigenvector, given an estimate of an eigenvalue.

  1: Pick a starting vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$, such that $\|\mathbf{x}^{(0)}\| = 1$;
  2: Let $\lambda^{(0)} = r(\mathbf{x}^{(0)}) = (\mathbf{x}^{(0)})^T A\mathbf{x}^{(0)}$;
  3: For $k = 1, 2, \cdots$;
  4: Solve $(A - \lambda^{(k-1)} I)\mathbf{v} = \mathbf{x}^{(k-1)}$ for $\mathbf{v}$;
  5: Let $\mathbf{x}^{(k)} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$;
  6: Let $\lambda^{(k)} = r(\mathbf{x}^{(k)}) = (\mathbf{x}^{(k)})^T A\mathbf{x}^{(k)}$;

- The method is only guaranteed to converge when the matrix A is both real and symmetric, and is known to fail in the cases where the matrix is not symmetric.

Why does Rayleigh Quotient work?

A good way to understand Rayleigh quotient iteration is as a sort of Newton iteration for the eigenvalue equations.

Why does Rayleigh Quotient work?

A good way to understand Rayleigh quotient iteration is as a sort of Newton iteration for the eigenvalue equations.

- Write

$$F(\mathbf{v}, \lambda) = A\mathbf{v} - \lambda\mathbf{v} = 0,$$

Why does Rayleigh Quotient work?

A good way to understand Rayleigh quotient iteration is as a sort of Newton iteration for the eigenvalue equations.

- Write
$$F(\mathbf{v}, \lambda) = A\mathbf{v} - \lambda\mathbf{v} = 0,$$
and differentiate to find
$$\triangle F = (A - \lambda I)\triangle\mathbf{v} - (\triangle\lambda)\mathbf{v}, \text{ Recall: } \triangle z = \frac{\partial z}{\partial x}\triangle x + \frac{\partial z}{\partial y}\triangle y.$$

Why does Rayleigh Quotient work?

A good way to understand Rayleigh quotient iteration is as a sort of Newton iteration for the eigenvalue equations.

- Write
$$F(\mathbf{v}, \lambda) = A\mathbf{v} - \lambda\mathbf{v} = 0,$$
and differentiate to find
$$\triangle F = (A - \lambda I)\triangle\mathbf{v} - (\triangle\lambda)\mathbf{v}, \text{ Recall: } \triangle z = \frac{\partial z}{\partial x}\triangle x + \frac{\partial z}{\partial y}\triangle y.$$
- Newton iteration gives
$$\begin{aligned} 0 &= F(\mathbf{v}_k, \lambda_k) + \triangle F(\mathbf{v}_k, \lambda_k) \\ &= (A - \lambda_k I)(\mathbf{v}_k + \triangle\mathbf{v}_k) - (\triangle\lambda_k)\mathbf{v}_k \\ &= (A - \lambda_k I)\mathbf{v}_{k+1} - (\triangle\lambda_k)\mathbf{v}_k, \end{aligned}$$
which means that $\mathbf{v}_{k+1} = (\triangle\lambda_k)(A - \lambda_k I)^{-1}\mathbf{v}_k$.

How to compute the corresponding eigenvalue?

Definition of Rayleigh quotient

For a real matrix A, the Rayleigh quotient of A, denoted as $R_A(\cdot)$ is a function from $R^n \backslash \{0\}$ to R, defined as follows

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

How to compute the corresponding eigenvalue?

Definition of Rayleigh quotient

For a real matrix A, the Rayleigh quotient of A, denoted as $R_A(\cdot)$ is a function from $R^n\backslash\{0\}$ to R, defined as follows

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}.$$

Also, for any $x \in R^n\backslash\{0\}$, we can express it in the orthonormal basis given by the eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_n$ of A, i.e.,

$$\mathbf{x}^T\mathbf{x} = \Big(\sum_{i=1}^{n} c_i\mathbf{v}_i\Big)^T\Big(\sum_{i=1}^{n} c_i\mathbf{v}_i\Big) = \sum_{i=1}^{n} c_i^2,$$

How to compute the corresponding eigenvalue?

Definition of Rayleigh quotient

For a real matrix A, the Rayleigh quotient of A, denoted as $R_A(\cdot)$ is a function from $R^n\backslash\{0\}$ to R, defined as follows

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}.$$

Also, for any $x \in R^n\backslash\{0\}$, we can express it in the orthonormal basis given by the eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_n$ of A, i.e.,

$$\mathbf{x}^T\mathbf{x} = \Big(\sum_{i=1}^{n} c_i\mathbf{v}_i\Big)^T\Big(\sum_{i=1}^{n} c_i\mathbf{v}_i\Big) = \sum_{i=1}^{n} c_i^2,$$

and

$$\mathbf{x}^T A\mathbf{x} = \Big(\sum_{i=1}^{n} c_i\mathbf{v}_i\Big)^T\Big(\sum_{i=1}^{n} c_i A\mathbf{v}_i\Big) = \sum_{i=1}^{n} c_i^2\lambda_i.$$

Rayleigh Quotient Cont'd

Now we can express $R_A(\mathbf{x})$ as

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\sum_{i=1}^{n} c_i^2 \lambda_i}{\sum_{i=1}^{n} c_i^2}.$$

Rayleigh Quotient Cont'd

Now we can express $R_A(\mathbf{x})$ as

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\sum_{i=1}^{n} c_i^2 \lambda_i}{\sum_{i=1}^{n} c_i^2}.$$

- Thus, we have

$$\lambda_1 = \max_{R^n \setminus \{0\}} R_A(\mathbf{x}), \text{ and } \lambda_n = \min_{R^n \setminus \{0\}} R_A(\mathbf{x}).$$

Rayleigh Quotient Cont'd

Now we can express $R_A(\mathbf{x})$ as

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\sum_{i=1}^n c_i^2 \lambda_i}{\sum_{i=1}^n c_i^2}.$$

- Thus, we have

$$\lambda_1 = \max_{R^n \setminus \{0\}} R_A(\mathbf{x}), \text{ and } \lambda_n = \min_{R^n \setminus \{0\}} R_A(\mathbf{x}).$$

- First note that if $\mathbf{v}_i$ is the eigenvector corresponding to the eigenvalue λ_i, then $R_A(\mathbf{v}_i) = \lambda_i$.

Analysis of Rayleigh Quotient method

Definition

Following the previous analysis, considering only the two largest eigenvalues,

$$R_A(\mathbf{x}^{(k)}) = \frac{{\mathbf{x}^{(k)}}^T A \mathbf{x}^{(k)}}{{\mathbf{x}^{(k)}}^T \mathbf{x}^{(k)}} \approx \frac{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^{k+1}\mathbf{v}_1 + c_2\lambda_2^{k+1}\mathbf{v}_2\right)}{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)}$$

$$= \frac{c_1^2\lambda_1^{2k+1} + c_2^2\lambda_2^{2k+1}}{c_1^2\lambda_1^{2k} + c_2^2\lambda_2^{2k}} = \lambda_1 \frac{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1}}{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}}$$

Analysis of Rayleigh Quotient method

Definition

Following the previous analysis, considering only the two largest eigenvalues,

$$R_A(\mathbf{x}^{(k)}) = \frac{{\mathbf{x}^{(k)}}^T A \mathbf{x}^{(k)}}{{\mathbf{x}^{(k)}}^T \mathbf{x}^{(k)}} \approx \frac{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^{k+1}\mathbf{v}_1 + c_2\lambda_2^{k+1}\mathbf{v}_2\right)}{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)}$$

$$= \frac{c_1^2\lambda_1^{2k+1} + c_2^2\lambda_2^{2k+1}}{c_1^2\lambda_1^{2k} + c_2^2\lambda_2^{2k}} = \lambda_1 \frac{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1}}{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}}$$

- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}$.

Analysis of Rayleigh Quotient method

Definition

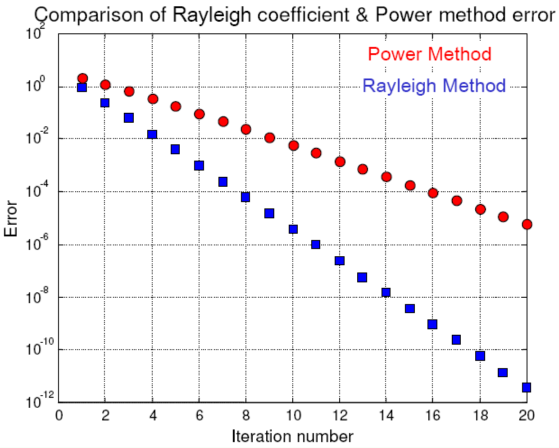
Following the previous analysis, considering only the two largest eigenvalues,

$$R_A(\mathbf{x}^{(k)}) = \frac{\mathbf{x}^{(k)\,T} A \mathbf{x}^{(k)}}{\mathbf{x}^{(k)\,T} \mathbf{x}^{(k)}} \approx \frac{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^{k+1} \mathbf{v}_1 + c_2\lambda_2^{k+1} \mathbf{v}_2\right)}{\left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)^T \left(c_1\lambda_1^k \mathbf{v}_1 + c_2\lambda_2^k \mathbf{v}_2\right)}$$

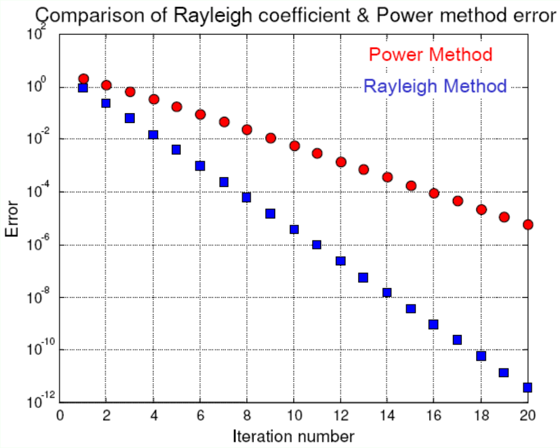
$$= \frac{c_1^2\lambda_1^{2k+1} + c_2^2\lambda_2^{2k+1}}{c_1^2\lambda_1^{2k} + c_2^2\lambda_2^{2k}} = \lambda_1 \frac{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1}}{1 + \frac{c_2^2}{c_1^2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}}$$

- The proportional error thus decays with successive iterations as $\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}$.
- It gives us a locally quadratically convergent algorithm, i.e. $\left|\frac{\lambda_2}{\lambda_1}\right|^{2}$.

Performance comparison



Performance comparison



Since the reduction per iteration is quadratic, that is, quadratic or second order convergence and is much faster

Outline

Motivation of deflation technique

So far we have considered how to find the largest eigenvalue. Suppose we have done this, then how do we find the rest?

Motivation of deflation technique

So far we have considered how to find the largest eigenvalue. Suppose we have done this, then how do we find the rest?
After finding the largest eigenvalue λ_1, a more general way forward is to make it into the smallest by deflation and then go on to find the new largest one, say λ_2.

Motivation of deflation technique

So far we have considered how to find the largest eigenvalue. Suppose we have done this, then how do we find the rest?

After finding the largest eigenvalue λ_1, a more general way forward is to make it into the smallest by deflation and then go on to find the new largest one, say λ_2.

We can repeatedly find all eigenvalues of a symmetric matrix A with distinct eigenvalues because

$$A \doteq A^{(0)} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T,$$

Motivation of deflation technique

So far we have considered how to find the largest eigenvalue. Suppose we have done this, then how do we find the rest?

After finding the largest eigenvalue λ_1, a more general way forward is to make it into the smallest by deflation and then go on to find the new largest one, say λ_2.

We can repeatedly find all eigenvalues of a symmetric matrix A with distinct eigenvalues because

$$A \doteq A^{(0)} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T,$$

and

$$A^{(i)} = A^{(i-1)} - \lambda_i \mathbf{v}_i \mathbf{v}_i^T.$$

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

We have

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = \begin{cases} 0 \cdot \mathbf{v}_j, & \text{if } j = 1; \\ \lambda_j \cdot \mathbf{v}_j, & \text{otherwise.} \end{cases}$$

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

We have

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = \begin{cases} 0 \cdot \mathbf{v}_j, & \text{if } j = 1; \\ \lambda_j \cdot \mathbf{v}_j, & \text{otherwise.} \end{cases}$$

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

We have

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = \begin{cases} 0 \cdot \mathbf{v}_j, & \text{if } j = 1; \\ \lambda_j \cdot \mathbf{v}_j, & \text{otherwise.} \end{cases}$$

- Thus, $A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T$ has the same eigenvectors as A, and the same eigenvalues as A except that the largest one has been replaced by 0.

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

We have

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = \begin{cases} 0 \cdot \mathbf{v}_j, & \text{if } j = 1; \\ \lambda_j \cdot \mathbf{v}_j, & \text{otherwise.} \end{cases}$$

- Thus, $A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T$ has the same eigenvectors as A, and the same eigenvalues as A except that the largest one has been replaced by 0.
- We can find the similar conclusion for $A^{(i-1)} - \lambda_i \mathbf{v}_i \mathbf{v}_i^T$.

Analysis of deflation technique

Consider

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = A\mathbf{v}_j - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T \mathbf{v}_j.$$

We have

$$(A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T)\mathbf{v}_j = \begin{cases} 0 \cdot \mathbf{v}_j, & \text{if } j = 1; \\ \lambda_j \cdot \mathbf{v}_j, & \text{otherwise.} \end{cases}$$

- Thus, $A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T$ has the same eigenvectors as A, and the same eigenvalues as A except that the largest one has been replaced by 0.
- We can find the similar conclusion for $A^{(i-1)} - \lambda_i \mathbf{v}_i \mathbf{v}_i^T$.
- Thus, we can use the power method with Rayleigh's coefficient to find the next biggest and so on.

Take-home messages

- Introduction to Eigenvalue and Eigenvector
- Calculating the Eigenvalue and Eigenvector
  - Power Method
  - Rayleigh Quotient Method
  - Deflation Technique