Lec 7: Eigenvectors

15-369/669/769: Numerical Computing

Instructor: Minchen Li

Table of Content

Table of Content

Motivation of the Eigenvalue Problem

Nonlinearity and Optimization Form

- We turn our attention now to a nonlinear problem about matrices: Finding their eigenvalues and eigenvectors, i.e. solving $Ax = \lambda x$.

- To see it is nonlinear, there is a product of unknowns λ and x.

  - Furthermore, to avoid the trivial solution $x = 0$, we constrain $\|\mathbf{x}\|_2 = 1$; keeping x on the unit sphere, which is not a vector space.

- Due to this structure, algorithms for finding eigenspaces will be considerably different from solving and analyzing linear systems.

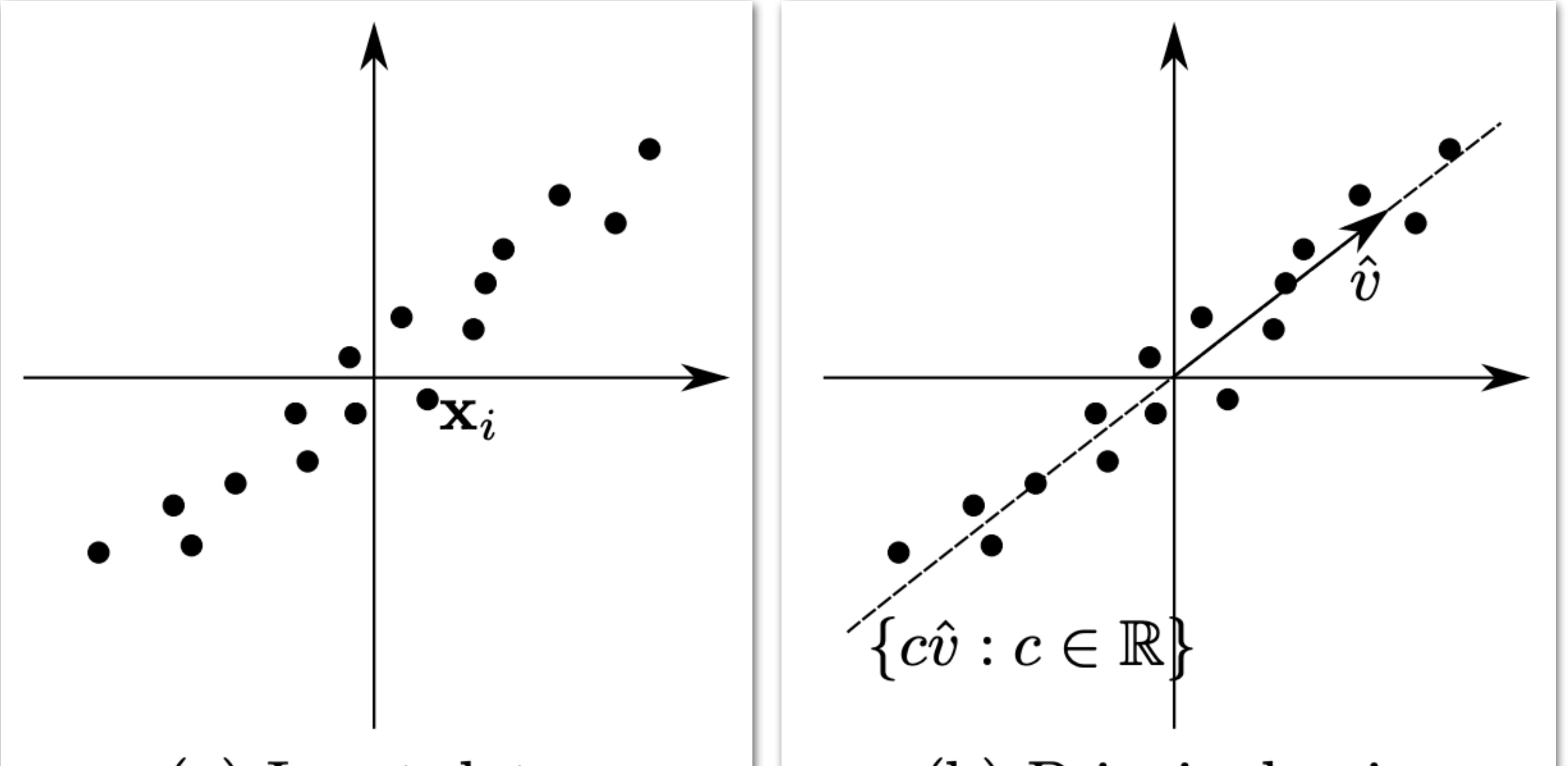
When A is symmetric, the eigenvectors of A are the critical points of $\mathbf{x}^\top A\mathbf{x}$ under the constraint $\|\mathbf{x}\|_2 = 1$.

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^T A\mathbf{x}) - \lambda \frac{\partial}{\partial \mathbf{x}}(\|\mathbf{x}\|^2 - 1) = 0$$

$$\Rightarrow A\mathbf{x} = \lambda \mathbf{x}$$

Motivation of the Eigenvalue Problem

An Example in Statistics



(a) Input data (b) Principal axis

- In a medical study, we may collect the age, weight, blood pressure, and heart rate of patients.

- Each patient i can be represented by a point $\mathbf{x}_i \in \mathbb{R}^4$ storing these 4 values.

- These statistics may exhibit strong correlations between the different dimensions, e.g., patients with higher blood pressures may be likely to have higher weights or heart rates.

- Thus, in reality the data may approximately live in a lower-dimensional space capturing the relationships between the different dimensions.

- Suppose there exists a 1-dimensional space approximating our dataset, we expect that there exists a vector $\mathbf{v}$ such that each data point $\mathbf{x}_i$ can be written as $\mathbf{x}_i = c_i\mathbf{v}$ for a different $c_i \in \mathbb{R}$.

Motivation of the Eigenvalue Problem

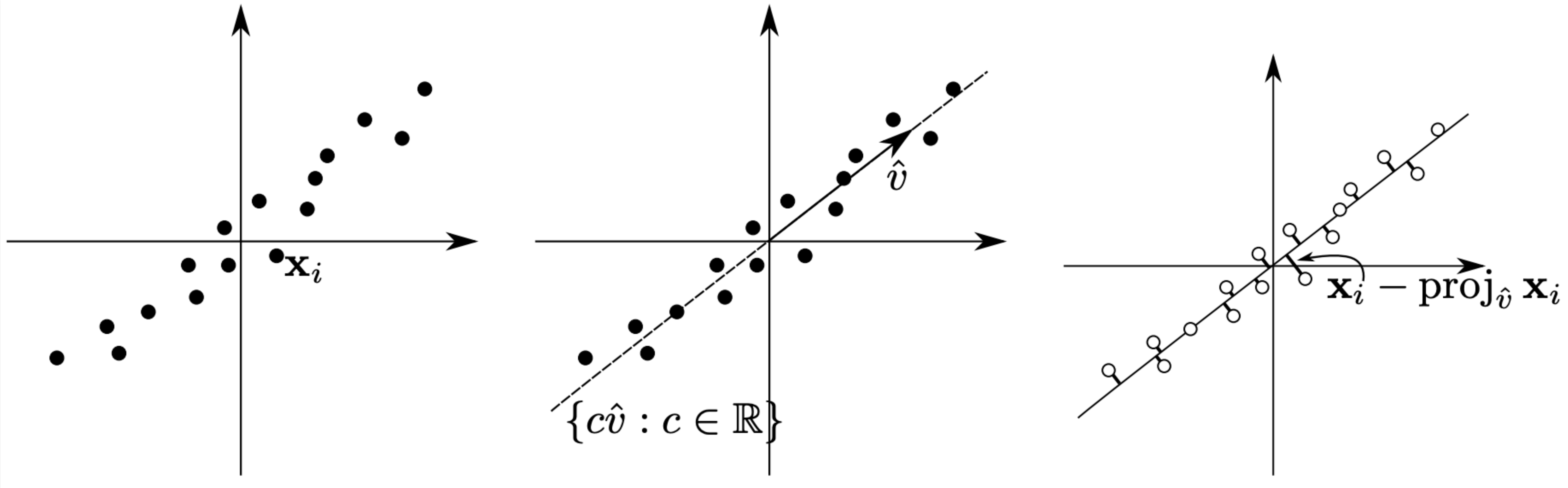
An Example in Statistics, Formulation

- The best approximation of $\mathbf{x}_i$ parallel to $\mathbf{v}$ is $\text{proj}_{\mathbf{v}}\mathbf{x}_i$. Let $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|_2$, we have:

$$\begin{aligned}\text{proj}_{\mathbf{v}}\,\mathbf{x}_i &= \frac{\mathbf{x}_i \cdot \mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\mathbf{v} \text{ by definition}\\ &= (\mathbf{x}_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} \text{ since } \mathbf{v}\cdot\mathbf{v} = \|\mathbf{v}\|_2^2.\end{aligned}$$

- Since $\|\mathbf{v}\|$ does not matter, we can restrict our search to the space of unit vectors $\hat{\mathbf{v}}$:

$$\begin{aligned}&\text{minimize}_{\hat{\mathbf{v}}} \; \sum_i \|\mathbf{x}_i - \text{proj}_{\hat{\mathbf{v}}}\,\mathbf{x}_i\|_2^2\\ &\text{subject to } \|\hat{\mathbf{v}}\|_2 = 1.\end{aligned}$$



(a) Input data (b) Principal axis (c) Projection error

Motivation of the Eigenvalue Problem

An Example in Statistics, Derivation and Solution

$$\begin{aligned}
\sum_i \|\mathbf{x}_i - \mathrm{proj}_{\hat{\mathbf{v}}}\, \mathbf{x}_i\|_2^2 &= \sum_i \|\mathbf{x}_i - (\mathbf{x}_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}}\|_2^2 \text{ as explained above} \\
&= \sum_i \left(\|\mathbf{x}_i\|_2^2 - 2(\mathbf{x}_i \cdot \hat{\mathbf{v}})(\mathbf{x}_i \cdot \hat{\mathbf{v}}) + (\mathbf{x}_i \cdot \hat{\mathbf{v}})^2 \|\hat{\mathbf{v}}\|_2^2\right) \text{ since } \|\mathbf{w}\|_2^2 = \mathbf{w} \cdot \mathbf{w} \\
&= \sum_i \left(\|\mathbf{x}_i\|_2^2 - (\mathbf{x}_i \cdot \hat{\mathbf{v}})^2\right) \text{ since } \|\hat{\mathbf{v}}\|_2 = 1 \\
&= \text{const.} - \sum_i (\mathbf{x}_i \cdot \hat{\mathbf{v}})^2 \text{ since the unknown here is } \hat{\mathbf{v}} \\
&= \text{const.} - \|X^\top \hat{\mathbf{v}}\|_2^2, \text{ where the columns of } X \text{ are the vectors } \mathbf{x}_i.
\end{aligned}$$

$$\boxed{\begin{array}{l} \text{maximize}_{\hat{\mathbf{v}}} \ \|X^\top \hat{\mathbf{v}}\|_2^2 \\ \text{subject to } \|\hat{\mathbf{v}}\|_2^2 = 1. \end{array}}$$

- Minimizing approximation error ⇔ Maximizing variance.
- We know $\|X^T\hat{\mathbf{v}}\|^2 = \hat{\mathbf{v}}^T X X^T \hat{\mathbf{v}}$, thus, $\hat{\mathbf{v}}$ is the eigenvector of XX^T with the highest eigenvalue. The vector $\hat{\mathbf{v}}$ is known as the **first principal component** of the dataset.

Table of Content

Properties of Eigenvectors

Definition

Definition 6.1 (Eigenvalue and eigenvector)**.** An *eigenvector* $\mathbf{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ of a matrix $A \in \mathbb{R}^{n \times n}$ is any vector satisfying $A\mathbf{x} = \lambda\mathbf{x}$ for some $\lambda \in \mathbb{R}$; the corresponding λ is known as an *eigenvalue*. *Complex* eigenvalues and eigenvectors satisfy the same relationships with $\lambda \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{C}^n$.

Definition 6.2 (Spectrum and spectral radius)**.** The *spectrum* of A is the set of eigenvalues of A. The *spectral radius* $\rho(A)$ is the maximum value $|\lambda|$ over all eigenvalues λ of A.

- The scale of an eigenvector is not important, since $A(c\mathbf{x}) = cA\mathbf{x} = c\lambda\mathbf{x} = \lambda(c\mathbf{x})$, so $c\mathbf{x}$ is an eigenvector with the same eigenvalue.

- Thus, we can restrict our search to those eigenvectors $\mathbf{x}$ with $\|\mathbf{x}\|^2 = 1$. But still, $\pm\mathbf{x}$ are both eigenvectors with the same eigenvalue.

Properties of Eigenvectors

Existence, Linear Dependency, and Diagonalizability

> **Proposition 6.1** ([5], Theorem 2.1)**.** Every matrix $A \in \mathbb{R}^{n \times n}$ has at least one (potentially complex) eigenvector.

> **Proposition 6.2** ([5], Proposition 2.2)**.** Eigenvectors corresponding to different eigenvalues must be linearly independent.

- An $n \times n$ matrix can have at most n distinct eigenvalues.
- The maximum number of linearly independent eigenvectors corresponding to an eigenvalue λ is the **geometric multiplicity** of λ.
- A matrix that does not have exactly n linearly independent eigenvectors are called defective.

> **Definition 6.3** (Nondefective)**.** A matrix $A \in \mathbb{R}^{n \times n}$ is *nondefective* or *diagonalizable* if its eigenvectors span $\mathbb{R}^n$.

Properties of Eigenvectors

An Example of Defective Matrix

Example 6.1 (Defective matrix)**.** The matrix

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

has only one linearly independent eigenvector $(1, 0)$, with eigenvalue $\lambda = 2$.

To see this, suppose $\mathbf{x} = (u, v)$ is an eigenvector of A. Expanding the relationships $A\mathbf{x} = \lambda\mathbf{x}$ and $\|\mathbf{x}\|_2^2 = 1$, we find

$$\begin{aligned} 2u + v - \lambda u &= 0 \\ 2v - \lambda v &= 0 \\ u^2 + v^2 &= 1 \end{aligned}$$

Factoring the second expression, we find $(2 - \lambda)v = 0$, so either $\lambda = 2$ or $v = 0$. If $\lambda = 2$, however, the first expression shows $v = 0$ anyway. Hence, all eigenvectors (u, v) of A satisfy $v = 0$, leaving us with a unique eigenvector $\mathbf{x} = (1, 0)$, up to sign.

Properties of Eigenvectors

Similarity Transformation

- If a matrix is nondefective, then it has n eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n$ with corresponding (possibly nonunique) eigenvalues $\lambda_1, \ldots, \lambda_n$.

- Take $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, we have $AX = XD$: a "stacked" version of $A\mathbf{x}_i = \lambda \mathbf{x}_i$. Further, $D = X^{-1}AX$, meaning A is **diagonalized** by a **similarity transformation**.

> **Definition 6.4** (Similar matrices)**.** Two matrices A and B are *similar* if there exists an invertible matrix T with $B = T^{-1}AT$.

- Similar matrices have the same eigenvalues:

$$B\mathbf{x} = \lambda\mathbf{x} \quad \overset{B = T^{-1}AT}{\Longrightarrow} \quad T^{-1}AT\mathbf{x} = \lambda\mathbf{x} \quad \Longrightarrow \quad AT\mathbf{x} = \lambda T\mathbf{x} \quad \Longrightarrow$$

$T\mathbf{x}$ is an eigenvector of A with eigenvalue λ

We can apply all the similarity transformations we want to a matrix without modifying its set of eigenvalues.

Properties of Eigenvectors

Hermitian Matrix

- Our original definition of eigenvalues allows them to be complex even if A is a real matrix.

- However, in the symmetric case we do not need complex arithmetic. We first generalize symmetric matrices to matrices in $\mathbb{C}^{n \times n}$:

> **Definition 6.5** (Complex conjugate)**.** The *complex conjugate* of a number $z = a + bi \in \mathbb{C}$, where $a, b \in \mathbb{R}$, is $\bar{z} := a - bi$. The complex conjugate $\bar{A}$ of a matrix $A \in \mathbb{C}^{m \times n}$ is the matrix with elements $\bar{a}_{ij}$.

> **Definition 6.6** (Conjugate transpose)**.** The *conjugate transpose* of $A \in \mathbb{C}^{m \times n}$ is $A^H := \bar{A}^\top$.

> **Definition 6.7** (Hermitian matrix)**.** A matrix $A \in \mathbb{C}^{n \times n}$ is *Hermitian* if $A = A^H$.

Properties of Eigenvectors

Spectral Theorem

> **Proposition 6.3.** All eigenvalues of Hermitian matrices are real.

> **Proposition 6.4.** Eigenvectors corresponding to distinct eigenvalues of Hermitian matrices must be orthogonal.

> **Theorem 6.1** (Spectral Theorem)**.** Suppose $A \in \mathbb{C}^{n\times n}$ is Hermitian (if $A \in \mathbb{R}^{n\times n}$, suppose it is symmetric). Then, A has exactly n orthonormal eigenvectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{C}^n$ with—possibly repeated—eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ (if $A \in \mathbb{R}^{n\times n}$, then $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^n$). In other words, there exists an orthogonal matrix X of eigenvectors and diagonal matrix D of eigenvalues such that $D = X^\top AX$.

$$A^{-1} = XD^{-1}X^\top$$

> **Proposition 6.5.** All eigenvalues of positive definite matrices are positive.

Table of Content

Computing a Single Eigenvalue

Power Iterations, Motivation

- Assume $A \in \mathbb{R}^{n \times n}$ is symmetric with a full set of eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}^n \in \mathbb{R}^n$, sorted such that their corresponding eigenvalues satisfy $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$.

- Take an arbitrary vector $\mathbf{v} \in \mathbb{R}^n$ and write it in the $\mathbf{x}_i$ basis as $\mathbf{v} = c_1\mathbf{x}_1 + \ldots + c_n\mathbf{x}_n$,

$$
\begin{aligned}
A\mathbf{v} &= c_1 A\mathbf{x}_1 + \cdots + c_n A\mathbf{x}_n \\
&= c_1\lambda_1\mathbf{x}_1 + \cdots + c_n\lambda_n\mathbf{x}_n \text{ since } A\mathbf{x}_i = \lambda_i\mathbf{x}_i \\
&= \lambda_1\left(c_1\mathbf{x}_1 + \frac{\lambda_2}{\lambda_1}c_2\mathbf{x}_2 + \cdots + \frac{\lambda_n}{\lambda_1}c_n\mathbf{x}_n\right) \\
A^2\mathbf{v} &= \lambda_1^2\left(c_1\mathbf{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2\mathbf{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n\mathbf{x}_n\right) \\
&\vdots \\
A^k\mathbf{v} &= \lambda_1^k\left(c_1\mathbf{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k c_2\mathbf{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k c_n\mathbf{x}_n\right).
\end{aligned}
$$

Computing a Single Eigenvalue

Power Iterations

$$A^k \mathbf{v} = \lambda_1^k \left(c_1 \mathbf{x}_1 + \left(\frac{\lambda_2}{\lambda_1} \right)^k c_2 \mathbf{x}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1} \right)^k c_n \mathbf{x}_n \right).$$

- As $k \to \infty$, the ratio $(\lambda_i/\lambda_1)^k \to 0$ unless $\lambda_i = \pm \lambda_1$, since $|\lambda_1| = \max(|\lambda_1|, \ldots, |\lambda_n|)$.
- If $\mathbf{x}$ is the projection of $\mathbf{v}$ onto the space of eigenvectors with eigenvalues λ_1, then (when the absolute values $|\lambda_i|$ are unique) as $k \to \infty$, $A^k \mathbf{v} \approx \lambda_1^k \mathbf{x}$.

This argument leads to an exceedingly simple algorithm for computing a single eigenvector $\mathbf{x}_1$ of A corresponding to its largest-magnitude eigenvalue λ_1:

1. Take $\mathbf{v}_1 \in \mathbb{R}^n$ to be an arbitrary nonzero vector.
2. Iterate for increasing k: $\mathbf{v}_k = A\mathbf{v}_{k-1}$ until ${}^{\mathbf{v}_k}/_{\|\mathbf{v}_k\|_2} \approx \pm{}^{\mathbf{v}_{k-1}}/_{\|\mathbf{v}_{k-1}\|_2}$ up to some tolerance.

Computing a Single Eigenvalue

Power Iterations, Remarks

- Although we have not considered the defective case, Power Iterations is still guaranteed to converge.
- The method converges the most quickly when $|\lambda_2|/|\lambda_1| \ll 1$.
- This technique may fail if we accidentally choose $\mathbf{v}_1$ such that $c_1 = 0$, but it rarely happens.

```
function POWER-ITERATION(A)
  v ← ARBITRARY(n)
  for k ← 1, 2, 3, ...
    v ← Av
  return v
```

```
function NORMALIZED-ITERATION(A)
  v ← ARBITRARY(n)
  for k ← 1, 2, 3, ...
    w ← Av
    v ← w/||w||
  return v
```

- The method can also fail if λ and $-\lambda$ are both eigenvalues of A with the largest magnitude.
- If $|\lambda_1| > 1$, then $\|\mathbf{v}_k\| \to \infty$ as $k \to \infty$. Since scale doesn't matter, we can normalize $\mathbf{v}_k$ at each step. Any norms would work.

Computing a Single Eigenvalue

Inverse Iterations

- If $A\mathbf{x} = \lambda\mathbf{x}$, then $\mathbf{x} = \lambda A^{-1}\mathbf{x}$, and so $A^{-1}\mathbf{x} = (1/\lambda)\mathbf{x}$. Thus, $1/\lambda$ is an eigenvalue of A^{-1} with eigenvector $\mathbf{x}$, and the smallest-magnitude eigenvalue of A corresponds to the largest-magnitude eigenvector of A^{-1}.

```
function INVERSE-ITERATION(A)
    v ← ARBITRARY(n)
    for k ← 1, 2, 3, ...
        w ← A^{-1} v
        v ← w/‖w‖
    return v
```

```
function INVERSE-ITERATION-LU(A)
    v ← ARBITRARY(n)
    L, U ← LU-FACTORIZE(A)
    for k ← 1, 2, 3, ...
        y ← FORWARD-SUBSTITUTE(L, v)
        w ← BACK-SUBSTITUTE(U, y)
        v ← w/‖w‖
    return v
```

Computing a Single Eigenvalue

Shifted Inverse Iterations

- Suppose $\sigma \in \mathbb{R}$ is close to (but not exactly) an eigenvalue λ of an $n \times n$ symmetric matrix A.

- If the eigenvalues of A are $\lambda_1, \ldots, \lambda_n$, then the eigenvalues of $(A - \sigma I_{n\times n})^{-1}$ are $(\lambda_1 - \sigma)^{-1}$, …, $(\lambda_n - \sigma)^{-1}$.

- Since σ is close to λ, the difference $|\lambda - \sigma|$ is close to zero, and hence $(\lambda - \sigma)^{-1}$ is likely to be the dominant eigenvalue of $(A - \sigma I_{n\times n})^{-1}$.

- Thus, applying eigenvalue iteration to $(A - \sigma I_{n\times n})^{-1}$, a slight generalization of inverse iteration, allows us to find the eigenvalue of A closest to σ.

- However, $(A - \sigma I_{n\times n})^{-1}$ would be close to singular. But somewhat surprisingly, this possible issue does not substantially affect shifted inverse iteration.

Computing a Single Eigenvalue

Rayleigh Quotient Iterations, Motivation

- Recall that power iteration converges fastest when ratio $|\lambda_2/\lambda_1|$ is close to zero.
- For shifted inverse iteration applied to $A - \sigma I_{n\times n}$, the dominant eigenvalue $(\lambda - \sigma)^{-1}$ becomes huge in magnitude as $\sigma \to \lambda$, which also makes the algorithm converge faster.
- But, the role of power iteration is exactly to improve our eigenvalue estimate! Hence, we might construct an algorithm by alternating between two steps:
  - Carry out one or more iterations of shifted inverse iteration on $(A - \sigma I_{n\times n})^{-1}$ to refine an estimate of the eigenvalue closest to σ.
  - Update σ to our improved estimate.

Computing a Single Eigenvalue

Rayleigh Quotient Iterations

- Suppose we have a fixed guess of an eigenvector $\mathbf{x}$ of A. Minimizing $\|A\mathbf{x} - \sigma\mathbf{x}\|^2$ over possible eigenvalue estimates $\sigma \in \mathbb{R}$ yields a least-squares approximation given by

$$\sigma = \frac{\mathbf{x}^\top A\mathbf{x}}{\|\mathbf{x}\|_2^2}.$$

— The Rayleigh quotient

function RAYLEIGH-QUOTIENT-ITERATION(A, σ)
  $\mathbf{v} \leftarrow$ ARBITRARY(n)
  for $k \leftarrow 1, 2, 3, \ldots$
    $\mathbf{w} \leftarrow (A - \sigma I_{n\times n})^{-1}\mathbf{v}$
    $\mathbf{v} \leftarrow \mathbf{w}/\|\mathbf{w}\|$
    $\sigma \leftarrow \frac{\mathbf{v}^\top A\mathbf{v}}{\|\mathbf{v}\|_2^2}$
  return $\mathbf{v}$

- Compared to Shifted Inverse Iterations:
  - More expensive iteration ($A - \sigma I_{n\times n}$ is changing and cannot be pre-factorized) but converges faster.

Computing a Single Eigenvalue

Summary

- **Power Iterations:** compute the eigenvalue with the largest magnitude
- **Inverse Iterations:** compute the eigenvalue with the smallest magnitude
- **Shifted Inverse Iterations:** compute the eigenvalue closest to an input estimate σ
- **Rayleigh Quotient Iterations:** compute the eigenvalue closest to an input estimate σ, a faster-converging variant of the Shifted Inverse Iterations
    - But each iteration is more expensive, thus overall performance is problem dependent.

Table of Content

Finding Multiple Eigenvalues

Deflation, Motivation

- In power iterations, when the initial eigenvector guess $\mathbf{v}_1$ is orthogonal to the first eigenvector $\mathbf{x}_1$, no matter how many times we apply A, the result will still be orthogonal to $\mathbf{x}_1$.

- Suppose we find $\mathbf{x}_1$ and λ_1 via power iteration, we can then restart the iteration after projecting $\mathbf{x}_1$ out of the initial guess $\mathbf{v}_1$.

  - Since the eigenvectors of A are orthogonal, power iteration after this projection will recover its second-largest eigenvalue!

- Due to finite-precision arithmetic, applying A to a vector may inadvertently introduce a small component parallel to $\mathbf{x}_1$. We can avoid this by projecting in each iteration.

Finding Multiple Eigenvalues

Deflation

```
function PROJECTED-ITERATION(symmetric A,k)
   for ℓ ← 1, 2, ..., k
      v_ℓ ← ARBITRARY(n)
      for p ← 1, 2, 3, ...
         u ← v_ℓ − proj_{span{v_1,...,v_{ℓ−1}}} v_ℓ
         w ← Au
         v_ℓ ← w/‖w‖
   return v_1, ..., v_k
```

The inner loop of projected iteration is equivalent to power iteration on the matrix AP, where P projects out $\mathbf{v}_1, \ldots, \mathbf{v}_{\ell-1}$:

$$P\mathbf{x} = \mathbf{x} - \mathrm{proj}_{\mathrm{span}\,\{\mathbf{v}_1,\ldots,\mathbf{v}_{\ell-1}\}}\,\mathbf{x}.$$

AP has the same eigenvectors as A with eigenvalues $0, \ldots, 0, \lambda_\ell, \ldots, \lambda_n$.

Finding Multiple Eigenvalues

Deflation via Householder Transformations

- In the previous strategy, AP might not be symmetric even if A is symmetric, and the iteration can fail if A is defective.

- Suppose $A\mathbf{x}_1 = \lambda_1\mathbf{x}_1$ with $\|\mathbf{x}_1\| = 1$. Take H to be the Householder matrix such that $H\mathbf{x}_1 = \mathbf{e}_1$. Since similarity transforms do not affect the set of eigenvalues, HAH^T will have the same eigenvalues as A. When we multiply HAH^T by $\mathbf{e}_1$:

$$\begin{aligned} HAH^\top \mathbf{e}_1 &= HAH\mathbf{e}_1 \text{ since } H \text{ is symmetric} \\ &= HA\mathbf{x}_1 \text{ since } H\mathbf{x}_1 = \mathbf{e}_1 \text{ and } H^2 = I_{n\times n} \\ &= \lambda_1 H\mathbf{x}_1 \text{ since } A\mathbf{x}_1 = \lambda_1\mathbf{x}_1 \\ &= \lambda_1\mathbf{e}_1 \text{ by definition of } H. \end{aligned}$$

$$HAH^\top = \begin{pmatrix} \lambda_1 & \mathbf{b}^\top \\ \mathbf{0} & B \end{pmatrix}.$$

- The matrix $B \in \mathbb{R}^{(n-1)\times(n-1)}$ has eigenvalues λ2, . . . , λn. Recursively applying this and power iteration we can find all eigenvalues.

Finding Multiple Eigenvalues

QR Iterations, Motivation

- Deflation has the drawback that each eigenvector must be computed separately, which can be slow and can accumulate error.
- To find more than one eigenvector simultaneously, consider

$$A^k = A^{k-1} \cdot A = \begin{pmatrix} | & | & & | \\ A^{k-1}\mathbf{a}_1 & A^{k-1}\mathbf{a}_2 & \cdots & A^{k-1}\mathbf{a}_n \\ | & | & & | \end{pmatrix}.$$

- When $k \to \infty$, we know $A^{k-1}\mathbf{a}_1 \to b_1\mathbf{x}_1$. By the deflation method, if we initially projected $\mathbf{x}_1$ out of $\mathbf{a}_2$, then $A^{k-1}P_{\mathbf{x}_1}\mathbf{a}_2 \to b_2\mathbf{x}_2$, where $P_{\mathbf{x}_1} = I - \mathbf{x}_1\mathbf{x}_1^T$ is a projection matrix.
- Interestingly, $AP_{\mathbf{x}} = P_{\mathbf{x}}A$ if $\mathbf{x}$ is an eigenvector of A, which means $P_{\mathbf{x}_1}A^{k-1}\mathbf{a}_2 \to b_2\mathbf{x}_2$.
- Thus, if we QR factorize $A^k = QR$, the columns of $\mathbf{Q}$ will converge to the eigenvectors of A.

Finding Multiple Eigenvalues

QR Iterations

```
function QR-ITERATION(A ∈ ℝ^{n×n})
    for k ← 1, 2, 3, ...
        Q, R ← QR-FACTORIZE(A)
        A ← RQ
    return diag(R)
```

- However, directly factorizing A^k would result in numerical issues, since cond $A^k \approx (\text{cond } A)^k$.

- What if we first factorize $A = Q_1 R_1$, and then factorize $(R_1 Q_1) = Q_2 R_2$, …?

$$
\begin{aligned}
A^2 &= (Q_1 R_1)(Q_1 R_1) \\
&= Q_1 (R_1 Q_1) R_1 \text{ by regrouping} \\
&= Q_1 Q_2 R_2 R_1 \text{ since } A_2 = R_1 Q_1 = Q_2 R_2 \\
A^3 &= A^2 \cdot A \\
&= Q_1 Q_2 R_2 R_1 \cdot Q_1 R_1 \text{ by the previous step} \\
&= Q_1 Q_2 R_2 (Q_2 R_2) R_1 \text{ since } A_2 = R_1 Q_1 = Q_2 R_2 \\
&= Q_1 Q_2 Q_3 R_3 R_2 R_1 \text{ since } A_3 = R_2 Q_2 = Q_3 R_3 \\
&\ \vdots \\
A^k &= Q_1 Q_2 \cdots Q_k R_k R_{k-1} \cdots R_1 \text{ by induction.}
\end{aligned}
$$

- Grouping the Q_i variables and the R_i variables separately provides a QR factorization of A^k.

- In practice, we first tri-diagonalize A using Householder matrices in $O(n^3)$ time, and then apply the QR iterations.

Finding Multiple Eigenvalues

QR Iterations, Example

- Applying QR iterations on $A = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}$:

$$A_1 = \begin{pmatrix} 2.000 & 3.000 \\ 3.000 & 2.000 \end{pmatrix} = \underbrace{\begin{pmatrix} -0.555 & 0.832 \\ -0.832 & -0.555 \end{pmatrix}}_{Q_1} \underbrace{\begin{pmatrix} -3.606 & -3.328 \\ 0.000 & 1.387 \end{pmatrix}}_{R_1}$$

$$\implies A_2 = R_1 Q_1 = \begin{pmatrix} 4.769 & -1.154 \\ -1.154 & -0.769 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 4.769 & -1.154 \\ -1.154 & -0.769 \end{pmatrix} = \underbrace{\begin{pmatrix} -0.972 & -0.235 \\ 0.235 & -0.972 \end{pmatrix}}_{Q_2} \underbrace{\begin{pmatrix} -4.907 & 0.941 \\ 0.000 & 1.019 \end{pmatrix}}_{R_2}$$

$$\implies A_3 = R_2 Q_2 = \begin{pmatrix} 4.990 & 0.240 \\ 0.240 & -0.990 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 4.990 & 0.240 \\ 0.240 & -0.990 \end{pmatrix} = \underbrace{\begin{pmatrix} -0.999 & 0.048 \\ -0.048 & -0.999 \end{pmatrix}}_{Q_3} \underbrace{\begin{pmatrix} -4.996 & -0.192 \\ 0.000 & 1.001 \end{pmatrix}}_{R_3}$$

$$\implies A_4 = R_3 Q_3 = \begin{pmatrix} 5.000 & -0.048 \\ -0.048 & -1.000 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 5.000 & -0.048 \\ -0.048 & -1.000 \end{pmatrix} = \underbrace{\begin{pmatrix} -1.000 & -0.010 \\ 0.010 & -1.000 \end{pmatrix}}_{Q_4} \underbrace{\begin{pmatrix} -5.000 & 0.038 \\ 0.000 & 1.000 \end{pmatrix}}_{R_4}$$

$$\implies A_5 = R_4 Q_4 = \begin{pmatrix} 5.000 & 0.010 \\ 0.010 & -1.000 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} 5.000 & 0.010 \\ 0.010 & -1.000 \end{pmatrix} = \underbrace{\begin{pmatrix} -1.000 & 0.002 \\ -0.002 & -1.000 \end{pmatrix}}_{Q_5} \underbrace{\begin{pmatrix} -5.000 & -0.008 \\ 0.000 & 1.000 \end{pmatrix}}_{R_5}$$

$$\implies A_6 = R_5 Q_5 = \begin{pmatrix} 5.000 & -0.002 \\ -0.002 & -1.000 \end{pmatrix}$$

$$A_6 = \begin{pmatrix} 5.000 & -0.002 \\ -0.002 & -1.000 \end{pmatrix} = \underbrace{\begin{pmatrix} -1.000 & -0.000 \\ 0.000 & -1.000 \end{pmatrix}}_{Q_6} \underbrace{\begin{pmatrix} -5.000 & 0.002 \\ 0.000 & 1.000 \end{pmatrix}}_{R_6}$$

$$\implies A_7 = R_6 Q_6 = \begin{pmatrix} 5.000 & 0.000 \\ 0.000 & -1.000 \end{pmatrix}$$

Table of Content