ECON2125/4021/8013

Lecture 10

John Stachurski

Semester 1, 2015

Transpose

The **transpose** of $\mathbf{A}$ is the matrix $\mathbf{A}'$ defined by

$$\mathrm{col}_n(\mathbf{A}') = \mathrm{row}_n(\mathbf{A})$$

Examples. If

$$\mathbf{A} := \begin{pmatrix} 10 & 40 \\ 20 & 50 \\ 30 & 60 \end{pmatrix} \quad \text{then} \quad \mathbf{A}' = \begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{pmatrix}$$

If

$$\mathbf{B} := \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} \quad \text{then} \quad \mathbf{B}' := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

Fact. For conformable matrices $\mathbf{A}$ and $\mathbf{B}$, transposition satisfies

1. $(\mathbf{A}')' = \mathbf{A}$
2. $(\mathbf{A}\mathbf{B})' = \mathbf{B}'\mathbf{A}'$
3. $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$
4. $(c\mathbf{A})' = c\mathbf{A}'$ for any constant c

For each square matrix $\mathbf{A}$,

1. $\det(\mathbf{A}') = \det(\mathbf{A})$
2. If $\mathbf{A}$ is nonsingular then so is $\mathbf{A}'$, and $(\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$

```
In [1]: import numpy as np

In [2]: A = np.random.randn(2, 2)

In [3]: np.linalg.inv(A.transpose())
Out[3]:
array([[ 4.52767206, -1.83628665],
       [ 0.90504942,  1.5014984 ]])

In [4]: np.linalg.inv(A).transpose()
Out[4]:
array([[ 4.52767206, -1.83628665],
      [ 0.90504942,  1.5014984 ]])
```

A square matrix $\mathbf{A}$ is called **symmetric** if $\mathbf{A}' = \mathbf{A}$

Equivalent: $a_{nk} = a_{kn}$ for all n, k

Examples.

$$\mathbf{A} := \begin{pmatrix} 10 & 20 \\ 20 & 50 \end{pmatrix}, \qquad \mathbf{B} := \begin{pmatrix} 1 & 2 & 3 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \end{pmatrix}$$

Ex. For any matrix $\mathbf{A}$, show that $\mathbf{A}'\mathbf{A}$ and $\mathbf{A}\mathbf{A}'$ are always

1. well-defined (multiplication makes sense)
2. symmetric

The **trace** of a square matrix is defined by

$$\operatorname{trace}\begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} = \sum_{n=1}^{N} a_{nn}$$

Fact. $\operatorname{trace}(\mathbf{A}) = \operatorname{trace}(\mathbf{A}')$

Fact. If $\mathbf{A}$ and $\mathbf{B}$ are square matrices and $\alpha, \beta \in \mathbb{R}$, then

$$\operatorname{trace}(\alpha\mathbf{A} + \beta\mathbf{B}) = \alpha \operatorname{trace}(\mathbf{A}) + \beta \operatorname{trace}(\mathbf{B})$$

Fact. When conformable, $\operatorname{trace}(\mathbf{AB}) = \operatorname{trace}(\mathbf{BA})$

A square matrix $\mathbf{A}$ is called **idempotent** if $\mathbf{A}\mathbf{A} = \mathbf{A}$

Examples.

$$\mathbf{A} := \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{I} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The next result is often used in statistics / econometrics:

Fact. If $\mathbf{A}$ is idempotent, then $\operatorname{rank}(\mathbf{A}) = \operatorname{trace}(\mathbf{A})$

Diagonal Matrices

Consider a square $N \times N$ matrix $\mathbf{A}$

The N elements of the form a_{nn} are called the **principal diagonal**

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix}$$

A square matrix $\mathbf{D}$ is called **diagonal** if all entries off the principal diagonal are zero

$$\mathbf{D} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_N \end{pmatrix}$$

Often written as

$$\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$$

Incidentally, the same notation works in Python

```
In [1]: import numpy as np

In [2]: D = np.diag((2, 4, 6, 8, 10))

In [3]: D
Out[3]:
array([[ 2,  0,  0,  0,  0],
       [ 0,  4,  0,  0,  0],
       [ 0,  0,  6,  0,  0],
       [ 0,  0,  0,  8,  0],
       [ 0,  0,  0,  0, 10]])
```

Diagonal systems are very easy to solve

Example.

$$\begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

is equivalent to

$$\begin{aligned} d_1 x_1 &= b_1 \\ d_2 x_2 &= b_2 \\ d_3 x_3 &= b_3 \end{aligned}$$

Fact. If $\mathbf{C} = \text{diag}(c_1, \ldots, c_N)$ and $\mathbf{D} = \text{diag}(d_1, \ldots, d_N)$ then

1. $\mathbf{C} + \mathbf{D} = \text{diag}(c_1 + d_1, \ldots, c_N + d_N)$
2. $\mathbf{C}\mathbf{D} = \text{diag}(c_1 d_1, \ldots, c_N d_N)$
3. $\mathbf{D}^k = \text{diag}(d_1^k, \ldots, d_N^k)$ for any $k \in \mathbb{N}$
4. $d_n \geq 0$ for all $n \implies \mathbf{D}^{1/2}$ exists and equals
$$\text{diag}(\sqrt{d_1}, \ldots, \sqrt{d_N})$$
5. $d_n \neq 0$ for all $n \implies \mathbf{D}$ is nonsingular and
$$\mathbf{D}^{-1} = \text{diag}(d_1^{-1}, \ldots, d_N^{-1})$$

Proofs: Check 1 and 2 directly, other parts follow

```
In [1]: import numpy as np

In [2]: D = np.diag((2, 4, 10, 100))

In [3]: np.linalg.inv(D)
Out[3]:
array([[ 0.5 ,  0.  ,  0.  ,  0.  ],
       [ 0.  ,  0.25,  0.  ,  0.  ],
       [ 0.  ,  0.  ,  0.1 ,  0.  ],
       [ 0.  ,  0.  ,  0.  ,  0.01]])
```

A square matrix is called **lower triangular** if every element strictly above the principle diagonal is zero

Example.

$$\mathbf{L} := \begin{pmatrix} 1 & 0 & 0 \\ 2 & 5 & 0 \\ 3 & 6 & 1 \end{pmatrix}$$

A square matrix is called **upper triangular** if every element strictly below the principle diagonal is zero

Example.

$$\mathbf{U} := \begin{pmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 0 & 0 & 1 \end{pmatrix}$$

Called **triangular** if either upper or lower triangular

Associated linear equations also simple to solve

Example.

$$\begin{pmatrix} 4 & 0 & 0 \\ 2 & 5 & 0 \\ 3 & 6 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

becomes

$$\begin{gathered} 4x_1 = b_1 \\ 2x_1 + 5x_2 = b_2 \\ 3x_1 + 6x_2 + x_3 = b_3 \end{gathered}$$

Top equation involves only x_1, so can solve for it directly

Plug that value into second equation, solve out for x_2, etc.

Eigenvalues and Eigenvectors

Let $\mathbf{A}$ be $N \times N$

In general $\mathbf{A}$ maps $\mathbf{x}$ to some arbitrary new location $\mathbf{Ax}$

But sometimes $\mathbf{x}$ will only be scaled:

$$\mathbf{Ax} = \lambda \mathbf{x} \quad \text{for some scalar } \lambda \tag{1}$$

If (1) holds and $\mathbf{x}$ is nonzero, then

1. $\mathbf{x}$ is called an **eigenvector** of $\mathbf{A}$ and λ is called an **eigenvalue**
2. $(\mathbf{x}, \lambda)$ is called an **eigenpair**

Clearly $(\mathbf{x}, \lambda)$ is an eigenpair of $\mathbf{A} \implies (\alpha\mathbf{x}, \lambda)$ is an eigenpair of $\mathbf{A}$ for any nonzero α

Example. Let

$$\mathbf{A} := \begin{pmatrix} 1 & -1 \\ 3 & 5 \end{pmatrix}$$

Then

$$\lambda = 2 \quad \text{and} \quad \mathbf{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

form an eigenpair because $\mathbf{x} \neq \mathbf{0}$ and

$$\mathbf{Ax} = \begin{pmatrix} 1 & -1 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \lambda \mathbf{x}$$

Example.

```
In [3]: import numpy as np
In [4]: A = [[1, 2],
   ...:      [2, 1]]

In [5]: eigvals, eigvecs = np.linalg.eig(A)

In [6]: x = eigvecs[:,0]  # Let x = first eigenvector
In [7]: lm = eigvals[0]   # Let lm = first eigenvalue

In [8]: np.dot(A, x)      # Compute Ax
Out[8]: array([ 2.12132034,  2.12132034])
In [9]: lm * x            # Compute lm x
Out[9]: array([ 2.12132034,  2.12132034])
```

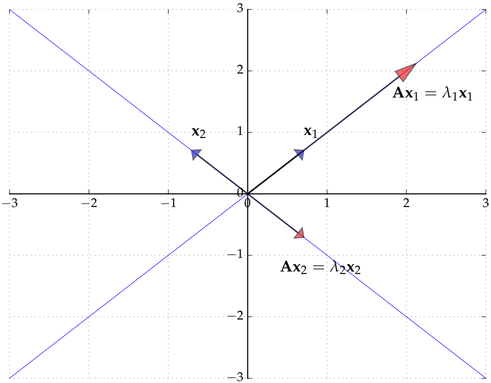


Figure : The eigenvectors of $\mathbf{A}$

Consider the matrix

$$\mathbf{R} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Induces counter-clockwise rotation on any point by 90°

Hence no point $\mathbf{x}$ is scaled

Hence there exists no pair $\lambda \in \mathbb{R}$ and $\mathbf{x} \neq \mathbf{0}$ such that

$$\mathbf{Rx} = \lambda \mathbf{x}$$

- In other words, no real-valued eigenpairs exist

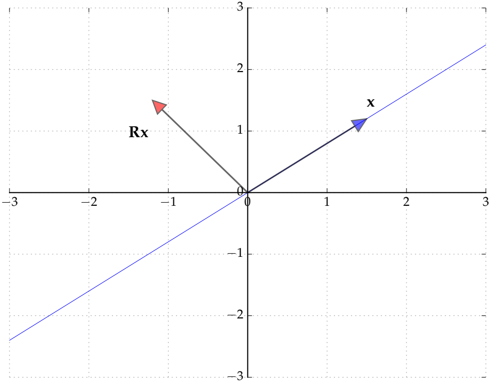


Figure : The matrix $\mathbf{R}$ rotates points by 90°

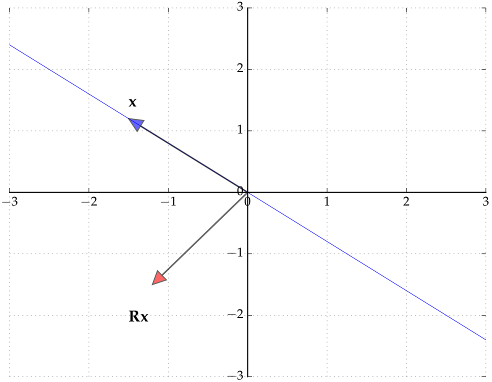


Figure : The matrix $\mathbf{R}$ rotates points by 90°

But $\mathbf{R}\mathbf{x} = \lambda\mathbf{x}$ can hold <u>if</u> we allow complex values

Example.

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -i \end{pmatrix} = \begin{pmatrix} i \\ 1 \end{pmatrix} = i \begin{pmatrix} 1 \\ -i \end{pmatrix}$$

That is,

$$\mathbf{R}\mathbf{x} = \lambda\mathbf{x} \quad \text{for} \quad \lambda := i \quad \text{and} \quad \mathbf{x} := \begin{pmatrix} 1 \\ -i \end{pmatrix}$$

Hence $(\mathbf{x}, \lambda)$ is an eigenpair provided we admit complex values

We do, since this is standard

Fact. For any square matrix $\mathbf{A}$

$$\lambda \text{ is an eigenvalue of } \mathbf{A} \iff \det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

Proof: Let $\mathbf{A}$ by $N \times N$ and let $\mathbf{I}$ be the $N \times N$ identity

We have

$$\begin{aligned}
\det(\mathbf{A} - \lambda \mathbf{I}) = 0 &\iff \mathbf{A} - \lambda \mathbf{I} \text{ is singular} \\
&\iff \exists\, \mathbf{x} \neq \mathbf{0} \text{ s.t. } (\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0} \\
&\iff \exists\, \mathbf{x} \neq \mathbf{0} \text{ s.t. } \mathbf{A}\mathbf{x} = \lambda \mathbf{x} \\
&\iff \lambda \text{ is an eigenvalue of } \mathbf{A}
\end{aligned}$$

Example. In the 2×2 case,

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \Longrightarrow \quad \mathbf{A} - \lambda \mathbf{I} = \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix}$$

$$\begin{aligned} \therefore \quad \det(\mathbf{A} - \lambda \mathbf{I}) &= (a - \lambda)(d - \lambda) - bc \\ &= \lambda^2 - (a + d)\lambda + (ad - bc) \end{aligned}$$

Hence the eigenvalues of $\mathbf{A}$ are given by the two roots of

$$\lambda^2 - (a + d)\lambda + (ad - bc) = 0$$

Equivalently,

$$\lambda^2 - \operatorname{trace}(\mathbf{A})\lambda + \det(\mathbf{A}) = 0$$

Existence of Eigenvalues

Fix $N \times N$ matrix $\mathbf{A}$

Fact. There exist complex numbers $\lambda_1, \ldots, \lambda_N$ such that

$$\det(\mathbf{A} - \lambda \mathbf{I}) = \prod_{n=1}^{N} (\lambda_n - \lambda)$$

Each such λ_i is an eigenvalue of $\mathbf{A}$ because

$$\det(\mathbf{A} - \lambda_i \mathbf{I}) = \prod_{n=1}^{N} (\lambda_n - \lambda_i) = 0$$

Important: Not all are necessarily distinct — there can be repeats

Fact. Given $N \times N$ matrix $\mathbf{A}$ with eigenvalues $\lambda_1, \ldots, \lambda_N$ we have

1. $\det(\mathbf{A}) = \prod_{n=1}^N \lambda_n$
2. $\operatorname{trace}(\mathbf{A}) = \sum_{n=1}^N \lambda_n$
3. If $\mathbf{A}$ is symmetric, then $\lambda_n \in \mathbb{R}$ for all n
4. If $\mathbf{A} = \operatorname{diag}(d_1, \ldots, d_N)$, then $\lambda_n = d_n$ for all n

Hence $\mathbf{A}$ is nonsingular $\iff$ all eigenvalues are nonzero (why?)

Fact. If $\mathbf{A}$ is nonsingular, then

$$\text{eigenvalues of } \mathbf{A}^{-1} = 1/\lambda_1, \ldots, 1/\lambda_N$$

Diagonalization

Square matrix $\mathbf{A}$ is said to be **similar** to square matrix $\mathbf{B}$ if

$$\exists \text{ invertible matrix } \mathbf{P} \text{ such that } \mathbf{A} = \mathbf{P}\mathbf{B}\mathbf{P}^{-1}$$

$$\begin{tikzcd}
\bullet \arrow[r, "\mathbf{B}"] & \bullet \arrow[d, "\mathbf{P}"] \\
\mathbf{x} \arrow[u, "\mathbf{P}^{-1}"] \arrow[r, "\mathbf{A}"'] & \mathbf{A}\mathbf{x}
\end{tikzcd}$$

Fact. If $\mathbf{A}$ is similar to $\mathbf{B}$, then $\mathbf{A}^t$ is similar to $\mathbf{B}^t$ for all $t \in \mathbb{N}$

Proof for case $t = 2$:

$$\begin{aligned}
\mathbf{A}^2 &= \mathbf{A}\mathbf{A} \\
&= \mathbf{P}\mathbf{B}\mathbf{P}^{-1}\mathbf{P}\mathbf{B}\mathbf{P}^{-1} \\
&= \mathbf{P}\mathbf{B}\mathbf{B}\mathbf{P}^{-1} \\
&= \mathbf{P}\mathbf{B}^2\mathbf{P}^{-1}
\end{aligned}$$

If $\mathbf{A}$ is similar to a diagonal matrix, then $\mathbf{A}$ is called **diagonalizable**

Fact. Let $\mathbf{A}$ be diagonalizable with $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ and let

1. $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$
2. $\mathbf{p}_n := \mathrm{col}_n(\mathbf{P})$

Then $(\mathbf{p}_n, \lambda_n)$ is an eigenpair of $\mathbf{A}$ for each n

Proof: From $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ we get $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{D}$

Equating n-th column on each side gives

$$\mathbf{A}\mathbf{p}_n = \lambda_n \mathbf{p}_n$$

Moreover $\mathbf{p}_n \neq \mathbf{0}$ because $\mathbf{P}$ is invertible (which facts?)

Fact. If $N \times N$ matrix $\mathbf{A}$ has N distinct eigenvalues $\lambda_1, \ldots, \lambda_N$, then $\mathbf{A}$ is diagonalizable as $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ where

1. $\mathbf{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$
2. $\operatorname{col}_n(\mathbf{P})$ is an eigenvector for λ_n

Example. Let

$$\mathbf{A} := \begin{pmatrix} 1 & -1 \\ 3 & 5 \end{pmatrix}$$

The eigenvalues of $\mathbf{A}$ are 2 and 4, while the eigenvectors are

$$\mathbf{p}_1 := \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \text{and} \quad \mathbf{p}_2 := \begin{pmatrix} 1 \\ -3 \end{pmatrix}$$

Hence

$$\mathbf{A} = \mathbf{P} \operatorname{diag}(2, 4) \mathbf{P}^{-1}$$

```
In [1]: import numpy as np
In [2]: from numpy.linalg import inv

In [3]: A = [[1, -1],
   ...:      [3, 5]]

In [4]: D = np.diag((2, 4))

In [5]: P = [[1, 1],  # Matrix of eigenvectors
   ...:     [-1, -3]]

In [6]: np.dot(P, np.dot(D, inv(P)))  # PDP^{-1} = A?
Out[6]:
array([[ 1., -1.],
       [ 3.,  5.]])
```

The Euclidean Matrix Norm

The concept of norm is very helpful for working with vectors

- provides notions of distance, similarity, convergence

How about an analogous concept for matrices?

Given $N \times K$ matrix $\mathbf{A}$, we define

$$\|\mathbf{A}\| := \max \left\{ \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \, : \, \mathbf{x} \in \mathbb{R}^K, \, \mathbf{x} \neq \mathbf{0} \right\}$$

- LHS is the **matrix norm** of $\mathbf{A}$
- RHS is ordinary Euclidean vector norms

In the maximization we can restrict attention to $\mathbf{x}$ s.t. $\|\mathbf{x}\| = 1$

To see this let

$$a := \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \qquad \text{and} \qquad b := \max_{\|\mathbf{x}\|=1} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|$$

Evidently $a \geq b$ because max is over a larger domain

To see the reverse let

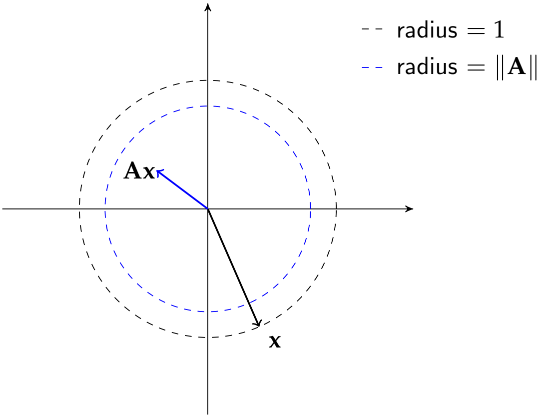
- $\mathbf{x}_a$ be the maximizer over $\mathbf{x} \neq \mathbf{0}$ and let $\alpha := 1/\|\mathbf{x}_a\|$
- $\mathbf{x}_b := \alpha\mathbf{x}_a$

Then

$$b \geq \frac{\|\mathbf{A}\mathbf{x}_b\|}{\|\mathbf{x}_b\|} = \frac{\|\alpha\mathbf{A}\mathbf{x}_a\|}{\|\alpha\mathbf{x}_a\|} = \frac{\alpha}{\alpha}\frac{\|\mathbf{A}\mathbf{x}_a\|}{\|\mathbf{x}_a\|} = a$$

Ex. Show that for any $\mathbf{x}$ we have $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$

If $\|\mathbf{A}\| < 1$ then $\mathbf{A}$ is called **contractive** — it shrinks the norm



The matrix norm has similar properties to the Euclidean norm

Fact. For conformable matrices $\mathbf{A}$ and $\mathbf{B}$, we have

1. $\|\mathbf{A}\| = \mathbf{0}$ if and only if all entries of $\mathbf{A}$ are zero
2. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$ for any scalar α
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$

The last inequality is called the submultiplicative property of the matrix norm

For square $\mathbf{A}$ it implies that $\|\mathbf{A}^k\| \leq \|\mathbf{A}\|^k$ for any $k \in \mathbb{N}$

Fact. For the diagonal matrix

$$\mathbf{D} = \operatorname{diag}(d_1, \ldots, d_N) = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_N \end{pmatrix}$$

we have

$$\|\mathbf{D}\| = \max_n |d_n|$$

Let $\{\mathbf{A}_j\}$ and $\mathbf{A}$ be $N \times K$ matrices

- If $\|\mathbf{A}_j - \mathbf{A}\| \to 0$ then we say that $\mathbf{A}_j$ **converges** to $\mathbf{A}$
- If $\sum_{j=1}^{J} \mathbf{A}_j$ converges to some matrix $\mathbf{B}_\infty$ as $J \to \infty$ we write

$$\sum_{j=1}^{\infty} \mathbf{A}_j = \mathbf{B}_\infty$$

In other words,

$$\mathbf{B}_\infty = \sum_{j=1}^{\infty} \mathbf{A}_j \quad \iff \quad \lim_{J \to \infty} \left\| \sum_{j=1}^{J} \mathbf{A}_j - \mathbf{B}_\infty \right\| \to 0$$

Neumann Series

Consider the difference equation $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{b}$, where

- $\mathbf{x}_t \in \mathbb{R}^N$ represents the values of some variables at time t
- $\mathbf{A}$ and $\mathbf{b}$ form the parameters in the law of motion for $\mathbf{x}_t$

Question of interest: is there an $\mathbf{x}$ such that

$$\mathbf{x}_t = \mathbf{x} \implies \mathbf{x}_{t+1} = \mathbf{x}$$

In other words, we seek an $\mathbf{x} \in \mathbb{R}^N$ that solves the system of equations

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \text{where} \quad \mathbf{A} \text{ is } N \times N \text{ and } \mathbf{b} \text{ is } N \times 1$$

We can get some insight from the scalar case $x = ax + b$

If $|a| < 1$, then this equation has the solution

$$\bar{x} = \frac{b}{1-a} = b \sum_{k=0}^{\infty} a^k$$

Does an analogous result hold in the vector case $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}$?

Yes, if we replace condition $|a| < 1$ with $\|\mathbf{A}\| < 1$

Let $\mathbf{b}$ be any vector in $\mathbb{R}^N$ and $\mathbf{A}$ be an $N \times N$ matrix

The next result is called the **Neumann series lemma**

Fact. If $\|\mathbf{A}^k\| < 1$ for some $k \in \mathbb{N}$, then $\mathbf{I} - \mathbf{A}$ is invertible and

$$(\mathbf{I} - \mathbf{A})^{-1} = \sum_{j=0}^{\infty} \mathbf{A}^j$$

In this case $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}$ has the unique solution

$$\bar{\mathbf{x}} = \sum_{j=0}^{\infty} \mathbf{A}^j \mathbf{b}$$

Sketch of proof that $(\mathbf{I}-\mathbf{A})^{-1}=\sum_{j=0}^{\infty}\mathbf{A}^j$ for case $\|\mathbf{A}\|<1$

We have $(\mathbf{I}-\mathbf{A})\sum_{j=0}^{\infty}\mathbf{A}^j=\mathbf{I}$ because

$$\begin{aligned}
\left\|(\mathbf{I}-\mathbf{A})\sum_{j=0}^{\infty}\mathbf{A}^j-\mathbf{I}\right\| &= \left\|(\mathbf{I}-\mathbf{A})\lim_{J\to\infty}\sum_{j=0}^{J}\mathbf{A}^j-\mathbf{I}\right\| \\
&= \lim_{J\to\infty}\left\|(\mathbf{I}-\mathbf{A})\sum_{j=0}^{J}\mathbf{A}^j-\mathbf{I}\right\| \\
&= \lim_{J\to\infty}\left\|\mathbf{A}^J\right\| \\
&\le \lim_{J\to\infty}\|\mathbf{A}\|^J=0
\end{aligned}$$

How to test the hypotheses of the Neumann series lemma?

The **spectral radius** of square matrix $\mathbf{A}$ is

$$\rho(\mathbf{A}) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathbf{A}\}$$

Here $|\lambda|$ is the **modulus** of the possibly complex number λ

Example. If $\lambda = a + ib$, then

$$|\lambda| = (a^2 + b^2)^{1/2}$$

Example. If $\lambda \in \mathbb{R}$, then $|\lambda|$ is the absolute value

Fact. If $\rho(\mathbf{A}) < 1$, then $\|\mathbf{A}^j\| < 1$ for some $j \in \mathbb{N}$

Proof, for diagonalizable $\mathbf{A}$:

We have $\mathbf{A}^j = \mathbf{P}\mathbf{D}^j\mathbf{P}^{-1}$ where

$$\mathbf{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N) \quad \text{and hence} \quad \mathbf{D}^j = \operatorname{diag}(\lambda_1^j, \ldots, \lambda_N^j)$$

Hence

$$\|\mathbf{A}^j\| = \|\mathbf{P}\mathbf{D}^j\mathbf{P}^{-1}\| \leq \|\mathbf{P}\|\|\mathbf{D}^j\|\|\mathbf{P}^{-1}\|$$

In particular, when $C := \|\mathbf{P}\|\|\mathbf{P}^{-1}\|$,

$$\|\mathbf{A}^j\| \leq C \max_n |\lambda_n^j| = C \max_n |\lambda_n|^j = C\rho(\mathbf{A})^j$$

This is < 1 for large enough j because $\rho(\mathbf{A}) < 1$