Lecture 19

The Power Method

( & FRIENDS )

* PSET 3 is out! Due 10/30 9PM.
  (has Julia component, so start early!)

* NEW checkup policy!
Last time: $A \in \mathbb{R}^{n \times n}$

Eigenvalue equation:

$Ax = \lambda x$

- **Matrix:** $A$
- **Scalar:** $\lambda$
- **Eigenvector:** $x$

$p(\lambda) = \det (A - \lambda I) = \prod_{i=1}^{n} (\lambda_i - \lambda)$

Sometimes

$A$ is diagonalizable

$A = T D T^{-1}$

- E.g., when all eigenvalues are distinct.
- More generally, if for every eigenvalue $\lambda_i$:
  - **Algebraic multiplicity:** $\text{mult.}(\lambda_i) = \text{geometric multiplicity.}(\lambda_i)$
  - $p(\lambda)$ has a factor $(\lambda_i - \lambda)^k$
  - $\dim \mathcal{N}(A-\lambda_i I) = k$
Assume $A \in \mathbb{R}^{n \times n}$ is diagonalizable, i.e.,

$$A = TDT^{-1}$$

Then $A^k = T D^k T^{-1}$

Polynomials applied to matrices

Let

$$q(t) = q_k t^k + q_{k-1} t^{k-1} + \ldots + q_1 t + q_0$$

be a univariate polynomial of degree $k$.

Then, we can define $q(A)$, where $A$ is a matrix.

$$q(A) = q_k A^k + q_{k-1} A^{k-1} + \ldots + q_1 A + q_0 I$$
In fact, this also makes sense for power series, if they converge.

Useful, for instance, to define the exponential of a matrix.

\[ e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}, \quad (x \in \mathbb{R}) \]

\[ e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}, \quad (A \in \mathbb{R}^{n \times n}) \]

(or sin/cos, log, \ldots)

Here's a nice (and useful) application:

Recall that:

\[ \frac{1}{1-x} = 1 + x + x^2 + \ldots + x^n + \ldots \quad \text{if } |x| < 1 \]

Then:

\[ (I-A)^{-1} = I + A + A^2 + A^3 + \ldots \]

(this converges if A is "small").
An interesting observation:

Let \( q(t) \) be a polynomial, and 

\[ A = TD T^{-1} \] 

a diagonalizable matrix.

Then

\[
q(A) = T \begin{bmatrix}
q(\lambda_1) & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & q(\lambda_n)
\end{bmatrix} T^{-1}
\]

Q: what happens if we choose \( q = \lambda \), i.e., what is

\[
p(A) = ?
\]
Example: \[ A = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \quad p(\lambda) = \lambda^2 - 3\lambda - 4 \]

Characteristic polynomial of \( A \).

\[ p(A) = A^2 - 3A - 4I \]
\[ = \begin{bmatrix} 7 & 6 \\ 9 & 10 \end{bmatrix} - \begin{bmatrix} 3 & 6 \\ 9 & 6 \end{bmatrix} - \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} = \]

In fact, this true for all matrices (even if they are not diagonalizable).

**Cayley-Hamilton Theorem:**

"Every square matrix satisfies its own characteristic polynomial," i.e., if \( p(\lambda) = \det(A - \lambda I) \) \( \Rightarrow \) \( p(A) = 0 \).
Diagonalization (and $A^k = T D^k T^{-1}$) also allows us to understand:

- The long-term behavior of linear dynamical systems.
- Asymptotic growth of recurrences.
- Convergence (and design) of algorithms.

(and much more!)
Example:

**Linear Dynamical Systems**

$X_{k+1} = AX_k$

$x_k$ is the state at time $k$

(e.g., vehicle, epidemic, population, algorithm, etc.)

Q: What is the long-term behavior of $x_k$, i.e., what happens as $k \to \infty$?

For instance:

- "constant"
- "unbounded oscillations"
- "exponential growth"
- "decay"
- "oscillatory decay"
- "sustained oscillation"

Q: How to decide??
Fibonacci's Rabbits

$f_0 = 0$
$f_1 = 1$
$f_2 = 1$
$f_3 = 2$
$f_4 = 3$

$f_{n+1} = f_n + f_{n-1}$

linear recurrence sequence (also, a dynamical system)
Define

\[
X_k = \begin{bmatrix} F_k \\ F_{k-1} \end{bmatrix} \Rightarrow \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_k \\ F_{k-1} \end{bmatrix}
\]

\[X_{k+1} = A \cdot X_k\]

**Q:** How does the sequence \( F_k \) behave for large \( k \)? (i.e., how many rabbits will we have in month \( k \)?)

**Diagonalization!**

\[p(\lambda) = \text{det} (A - \lambda I) = \lambda^2 - \lambda - 1\]

**Eigenvectors of** \( A \)

\[\lambda_1 = \phi, \quad \lambda_2 = 1 - \phi\]

\[\phi = \frac{1 + \sqrt{5}}{2} = 1.618...\]

"golden ratio"
\[ X_k = A^k X_0 \]

If \( A = T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} T^{-1} \)

\[ A^k = T \begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} T^{-1} \]

\[ \lambda_1 = \phi \approx 1.61 \]
\[ \lambda_2 = 1 - \phi \approx -0.61 \]

\[ \lambda_1^k \rightarrow \infty \quad \text{as} \ k \rightarrow \infty \]
\[ \lambda_2^k \rightarrow 0 \]

\[ \Rightarrow \text{For large } k, \quad F_k \approx \phi^k = (1.618\ldots)^k \]

"exponential growth"

rate is given by
largest eigenvalue \( (\phi) \)
Stability: \( X_{k+1} = A X_k \), fixed initial condition \( X_0 \)

Does the state \( X_k \) go to zero, as \( k \to \infty \)?

\[
A^k = T \begin{bmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{bmatrix} T^{-1}
\]

Thm: If all eigenvalues satisfy \( |\lambda_i| < 1 \),

then \( X_k \to 0 \) as \( k \to \infty \).

“System is stable if all eigenvalues are strictly inside the unit disk.”
Word asymptotics

How many "words" of length \( n \) (in the letters \( A, B \)) are there, that do not contain the string "ABA"?

E.g.: AABABAABABA vs BABAABAAA3

Hey, this about counting paths on a graph!
Q1: If I do not have the constraint, how many words of length $N$ are there?

$w(N) \sim 2^N$

Q2: With the constraint (no $ABA$), how many?

$w(N) \sim (1.7548)^N$ (for large $N$)

Largest eigenvalue (in absolute value) of $G$
POWER ITERATION

Natural method for computing one eigenvalue/eigenvector. $\lambda_1, v_1$

Many variations exist, here we'll present the simplest version.

- **Matrix** $A \in \mathbb{R}^{n \times n}$, eigs $\lambda_1, \ldots, \lambda_n$
- **Dominant eigenvalue**: $|\lambda_1| > |\lambda_i|, \quad i = 2, \ldots, n$
- **Initial vector** $x_0 \in \mathbb{R}^n$
  is "random"
(more precisely, $x_0$ has a nonzero component along $v_1$)
Iteration:

$$X_{k+1} = \frac{A \cdot X_k}{||A \cdot X_k||}$$

(multiply by $A$, then normalize)

Then, as $k \to \infty$

- $X_k$ converges to eigenvector $v_1$
- $X_k^T A X_k$ converges to eigenvalue $\lambda_1$

Why?

$$X_k \sim A^k x_0 = T D^k T^{-1} x_0$$

$$= T\begin{pmatrix} \lambda_1^k & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n^k \end{pmatrix} (T^{-1}x_0)$$

$$\approx T\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} (\ast)$$

$$\approx v_1$$

$T = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}$

$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$

First component is nonzero

(A2)

$|\lambda_i^k| \gg |\lambda_1^k|$ (A2)
What determines convergence rate? \[ |\lambda_1| > |\lambda_2| > |\lambda_3| \ldots \]

(Slow, fast)

- Spectral gap: \[ \frac{|\lambda_1|}{|\lambda_2|} \]

Power iteration is very useful for large, sparse matrices.

(These typically arise, e.g., from graphs or PDEs)

Variations

- Specific eigenvalues (inverse iteration, …)
- Faster convergence (Rayleigh quotient, …)