Math of Data Science: Lecture 2

Vlad Kobzar

APAM, Columbia

September 8, 2022

- Last time - course intro
- Today - linear algebra review: diagonalization, projections

Linear algebra - motivation

- Data is naturally represented by linear algebra objects
  - vectors represent, e.g., features of the data or biases of a neural network
  - matrices represent multiple observations of the features or weights of a neural network
- Understanding the structure of a matrix can reveal structure in the data, e.g, PCA
- Projections allow us to reduce dimensionality/denoise the data
- Numerical linear algebra allows us to perform matrix computations

Eigendecomposition

An **eigenvector** x of a square matrix A satisfies

$$Ax = \lambda x$$

for scalar λ which is the corresponding **eigenvalue**. Even if A is real, in general its eigenvectors and eigenvalues can be complex.

Eigendecomposition

If a square matrix $A \in \mathbb{R}^{n \times n}$ has n linearly independent eigenvectors $x_1, \ldots, x_n$ (with eigenvalues $\lambda_1, \ldots, \lambda_n$), it can be expressed in terms of a matrix X, whose columns are the eigenvectors, and a diagonal matrix containing the eigenvalues,

$$
\begin{aligned}
A &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^{-1} \\
&= X \Lambda X^{-1}
\end{aligned}
$$

Pf:

$$
\begin{aligned}
AX &= \begin{bmatrix} Ax_1 & Ax_2 & \cdots & Ax_n \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \end{bmatrix} \\
&= X \Lambda
\end{aligned}
$$

Example: computing matrix powers

Assume that we want to compute

$$AA \cdots Ax = A^k x, \tag{1}$$

If A has an eigendecomposition,

$$\begin{aligned} A^k &= X\Lambda X^{-1} X\Lambda X^{-1} \cdots X\Lambda X^{-1} \\ &= X\Lambda^k X^{-1} \\ &= X \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \lambda_2^k & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \lambda_n^k \end{bmatrix} X^{-1}, \end{aligned}$$

Computing eigenvalues

- In your linear algebra course, you probably computed eigenvalues by solving the characteristic polynomial

$$\det(A - \lambda I) = 0$$

- In practice, this is not feasible due to numerical stability issues
  - Let $g(\lambda) = \det A$, and note that $g'(\lambda) = \frac{\det A}{\lambda}$. Then a linear approximation of the determinant:

$$\Delta x = \det(A - (\lambda + \Delta\lambda)I) \approx \det A - (\lambda + \Delta\lambda)\frac{\det A}{\lambda}$$

  - Thus if we have a numerical error of Δx when we evaluate the characteristic polynomial, it translates into error

$$\Delta\lambda = -\frac{\lambda}{\det A}\Delta x$$

    for the particular eigenvalue, which is blows up if the other eigenvalues are small.

Computing eigenvectors - power method

- Let $A \in \mathbb{R}^{n \times n}$ be a matrix with eigendecomposition $X \Lambda X^{-1}$ and let v be an arbitrary vector in $\mathbb{R}^n$.
- Since the columns of X are linearly independent, they form a basis for $\mathbb{R}^n$, so

$$v = \sum_{i=1}^{n} c_i X_{:i}, \quad c_i \in \mathbb{R},\ 1 \leq i \leq n. \tag{2}$$

- Then,

$$A^k v = \sum_{i=1}^{n} c_i A^k X_{:i} = \sum_{i=1}^{n} c_i \lambda_i^k X_{:i}$$

- Assume that $|\lambda_1| > |\lambda_2| \geq \ldots$, and $c_1 \neq 0$ (the latter happens with probability 1 if we draw a random v)
- Then as k grows, the term $c_1 \lambda_1^k X_{:1}$ will dominate the other terms.

Power method

- $c_1 \lambda_1^k X_{:1} \to \infty$ or 0 unless we normalize before applying A.

Algorithm 1: Power method

Input: A matrix A.
Output: An estimate of the eigenvector of A corresponding to the largest eigenvalue.
Initialization: Set $v_1 := v / \left\|v\right\|_2$, where v contains random entries.
For $i = 1, \ldots, k$, compute

$$v_i := \frac{A v_{i-1}}{\|A v_{i-1}\|_2}.$$

- This method has been reportedly used in Google's PageRank algorithm and industrial recommendation systems
- Mainly used for non-symmetric matrices

Symmetric matrices

- $S \in \mathbb{R}^{n \times n}$ is *symmetric* if $S^\top = S$ (or equivalently $S_{ij} = S_{ji}$.
- These matrices arise naturally in data science
- If, for example, S_{ij} corresponds to some similarity measure, like covariance or distance between features i and j.

Symmetric matrices: eigendecomposition

If $S \in \mathbb{R}^{n \times n}$ is real symmetric, then it has an eigendecomposition of the form

$$S = Q \Lambda Q^T \tag{3}$$

where Λ is a real diagonal matrix and $Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}$ is an orthogonal matrix.

- It turns out that every $n \times n$ symmetric matrix has n linearly independent vectors.
- The proof of this fact is not very instructive, so we'll just assume it as true.
- Then we can show that the eigenvalues are real and the eigenvectors are real and orthonormal

Symmetric matrices: real eigenvalues

- The conjugate transpose of a complex vector is $x^* := \bar{x}^\top$, i.e., the imaginary part of each component of the transpose $x^\top$ of x is negated.
- One can see that $x^*x = \langle x, x \rangle = \|x\|_2^2$.
- Conjugation distributes over multiplication,e.g., $(\lambda x)^* = \bar{\lambda} x^*$
- Assuming an eigenvector x has norm 1

$$x^* S x = \lambda x^* x = \lambda$$

and at the same time

$$x^* S x = (Sx)^* x = (\lambda x)^* x = \bar{\lambda}$$

- Thus, $\lambda = \bar{\lambda}$ and therefore its imaginary part is zero

Symmetric matrices: real eigenvalues

- If an eigenvector is complex, then its real and/or imaginary parts $y, z \in \mathbb{R}^n$ are also eigenvector(s) to the extent they are nonzero

$$S(y + iz) = \lambda(y + iz) \to Sy = \lambda y, Sz = \lambda z$$

- And at least one of them must be nonzero since the complex eigenvector is nonzero

Symmetric matrices: eigenvectors are orthonormal

- If m linearly independent eigenvectors correspond to the same eigenvalue λ, then their linear combination is also an eigenvector corresponding to λ.
- Therefore, they can be orthonormalized by Gram-Schmidt (see p. 128 of Strang)
- The resulting orthonormal set will also be m linearly independent eigenvectors corresponding to λ

Symmetric matrices: eigenvectors are orthonormal

- If two eigenvectors correspond to different eigenvalues, first assume one of them is zero and the other λ is not:

$$Sx = \lambda x \text{ and } Sy = 0$$

- For any matrix A,the nullspace $\mathrm{N}(A)$ is orthogonal to the column space $C(A^T)$ of its transpose (see, e.g., p.31 of Strang)
- And for a symmetric matrix S, $C(S^T) = C(S)$
- Since $x \in C(S)$ and $y \in N(S)$, we have $x \perp y$.

Symmetric matrices: eigenvectors are orthonormal

- If two eigenvectors correspond to two different nonzero eigenvalues:
$$Sx = \lambda x \text{ and } Sy = \alpha y$$
then
$$(S - \alpha I)y = 0$$
and
$$(S - \alpha I)x = (\lambda - \alpha)x$$
for $\lambda - \alpha \neq 0$
- Since $x \in C(S - \alpha I)$ and $y \in N(S - \alpha I)$, we again have $x \perp y$.

Eigendecomposition of S as an optimization problem

- The eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ of a symmetric matrix S, determine the **quadratic form**:

$$f(x) := x^T S x = x^T Q \Lambda Q^T x = \sum_{i=1}^{n} \lambda_i \left(x^T q_i \right)^2 \qquad (4)$$

- λ_1 is the maximum attained by f if $\|x\|_2 = 1$
- λ_2 is the maximum if we restrict x to be normalized and orthogonal to the first eigenvector q_1, and so on.

Eigendecomposition of S as an optimization problem

Theorem

For any symmetric matrix $S \in \mathbb{R}^n$ with normalized eigenvectors $q_1, q_2, \ldots, q_n$ with corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$

$$\lambda_1 = \max_{||q||_2=1} q^T S q, \tag{5}$$

$$q_1 = \arg\max_{||q||_2=1} q^T S q, \tag{6}$$

$$\lambda_k = \max_{||q||_2=1, q \perp q_1, \ldots, q_{k-1}} q^T S q, \tag{7}$$

$$q_k = \arg\max_{||q||_2=1, q \perp q_1, \ldots, q_{k-1}} q^T S q. \tag{8}$$

Eigendecomposition of S as an optimization problem

- The eigenvectors are an orthonormal basis (they are mutually orthogonal and we assume that they have been normalized)
- so we can represent any unit-norm vector h_k that is orthogonal to $q_1, \ldots, q_{k-1}$ as

$$h_k = \sum_{i=k}^{n} \alpha_i q_i \tag{9}$$

where

$$||h_k||_2^2 = \sum_{i=k}^{n} \alpha_i^2 = 1, \tag{10}$$

Note that h_1 is just an arbitrary unit-norm vector.

Eigendecomposition of S as an optimization problem

- Now we will show that the value of $f(h_k)$ when the normalized h_k is restricted to be orthogonal to $q_1, \ldots, q_{k-1}$ cannot be larger than λ_k,

$$
\begin{aligned}
h_k^T S h_k &= \sum_{i=1}^{n} \lambda_i \Big(\sum_{j=k}^{m} \alpha_j q_i^T q_j\Big)^2 \quad \text{by (4) and (9)} \\
&= \sum_{i=1}^{n} \lambda_i \alpha_i^2 \quad \text{because } q_1, \ldots, q_m \text{ is an orthonormal basis} \\
&\leq \lambda_k \sum_{i=k}^{m} \alpha_i^2 \quad \text{because } \lambda_k \geq \lambda_{k+1} \geq \ldots \geq \lambda_m \\
&= \lambda_k, \quad \text{by (10).}
\end{aligned}
$$

Eigendecomposition of S as an optimization problem

- To prove the theorem we just need to show that q_k achieves the maximum:

$$\begin{aligned} q_k^T S q_k &= \sum_{i=1}^{n} \lambda_i (q_i^T q_k)^2 \\ &= \lambda_k. \end{aligned}$$

Projections - motivation

- Data is naturally represented by vectors and matrices
- Projections allow us to:
  - reduce dimensionality/denoise data;
  - use iterative optimization methods to minimize a function subject to constraints

Projections

- Any matrix $U \in \mathbb{R}^{k \times m}$ can be viewed as a "projection"
- It is linear transformation $U : \mathbb{R}^n \to \mathbb{R}^k$
- Any matrix $P \in \mathbb{R}^{m \times m}$ that satisfies $P^2 = P$ is called a *projection matrix*.
- It's image or $C(P)$ is a k-dimensional linear subspace of $\mathbb{R}^m$, e.g., a line, plane or hyperplane
  - A projection Π (satisfying $\Pi^2 = \Pi$) to a non-linear subset, e.g. l^2 unit ball, won't be given by a matrix

Orthogonal projections

- If $U \in \mathbb{R}^{k \times m}$ has orthonormal rows (can happen only if $k \leq m$), then $UU^T = I$
- $P = U^T U$ is a symmetric projection matrix

$$P^2 = (U^T U) U^T U = U^T I U = P$$

- The basis of the subspace is given by rows of U.
- Example $U = [\cos\theta \;\; \sin\theta]$.

Orthogonal projections

- Strang defines *orthogonal projection* as follows: "If

$$P^2 = P = P^T$$

  then Pb is the orthogonal projection of b on the column space of P."
- Would this definition be equivalent if $P = U^T U$ for some $U \in \mathbb{R}^{k \times m}$ with orthonormal rows instead of $P^T = P$? (One direction is shown on the previous page).

Orthogonal projections

- We can prove the other direction: i.e.,

$$P^2 = P \text{ and } P^T = P \Rightarrow P = U^T U$$

  for some $U \in \mathbb{R}^{k \times m}$ with orthonormal rows
  - $P^T = P$ implies that $P = V^T \Lambda V$ for an orthogonal $V \in \mathbb{R}^m$
  - $P^2 = (V^T \Lambda V) V^T \Lambda V = V^T \Lambda^2 V = V^T \Lambda V = P$,
  - This in turn implies that $\Lambda^2 = \Lambda$
  - Therefore, Λ can only have 0 and 1 entries on the diagonal
  - Take U to be V after removing the rows in the position corresponding to the zero eigenvalues
  - Then: $P = U^T U$

Projections

Theorem (Properties of orthogonal projections)

Every vector $x \in \mathbb{R}^m$ has a **unique** *orthogonal projection Px onto any subspace $\mathcal{S} \subseteq \mathbb{R}^m$ of finite dimension. In particular x can be expressed as*

$$x = Px + (I - P)x \tag{11}$$

- *One can prove that $(I - P)$ is also an orthogonal projection*
- *And it's a projection on the orthogonal complement $\mathcal{S}^\perp$*

Projections

- Assume $x_1' \in \mathcal{S}, x_2' \in \mathcal{S}^\perp$ such that $x = x_1' + x_2'$
- Since $(x_1 - x_1') + (x_2 - x_2') = 0$, $\|(x_1' - x_1) + (x_2 - x_2')\| = 0$
- Then $x_1 - x_1' \in \mathcal{S}$ and $x_2 - x_2' \in \mathcal{S}^\perp$ implies

$$\|(x_1' - x_1) + (x_2 - x_2')\|^2 = \|(x_1' - x_1)\|^2 + \|(x_2 - x_2')\|^2$$

- so the above expression is zero, i.e., orthogonal projection is unique.

Projections as optimization

Theorem

The orthogonal projection Px of a vector x onto a subspace $\mathcal{S}$ is the closest vector to x in the l^2 norm that belongs to $\mathcal{S}$ in , i.e. Px solves the optimization problem

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \|x - u\| \\ \text{subject to} \quad & u \in \mathcal{S}. \end{aligned}$$

Projections as optimization

Proof.

- Take any point $u \in \mathcal{S}$ such that $u \neq Px$

$$
\begin{aligned}
||x-u||^2 &= ||(I-P)x + Px - u||^2 && (12)\\
&= ||(I-P)x||^2 + ||Px-u||^2 + 2\left\langle (I-P)x, Px-u \right\rangle && (13)\\
&= ||(I-P)x||^2 + ||Px-u||^2 && (14)
\end{aligned}
$$

  where (14) follows because $(I-P)x$ belongs to $S^{\perp}$ and $Px-u$ to S.

- If $u \neq Px$, then $||Px-u||^2 > 0$.
- Therefore, the optimal $u = Px$.

$\square$

Next steps

- Finish review of linear algebra: SVD
- Review probability and optimization
- PCA

References I

[1] Strang, Linear Algebra and Learning from Data, Wellesley Cambridge Press, 2019 2012

[2] Carlos Fernandez-Granda, *DS-GA 1013 / MATH-GA 2821 Mathematical Tools for Data Science, Lecture Notes*, 2020

[3] Carlos Fernandez-Granda, *Probability and Statistics for Data Science, Lecture Notes*, 2017 `https://cims.nyu.edu/~cfgranda/pages/stuff/probability_stats_for_DS.pdf`