Lecture 8: Algorithms for PCA, Eigenvalues + Eigenvectors

Recall: $x_1, \dots, x_n \in \mathbb{R}^d$, $\frac{1}{n}\sum x_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$.

k-PCA objective:

$$\underset{\substack{v_1, \dots, v_k \\ \text{orthonormal}}}{\operatorname{argmax}} \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{k} \langle x_i, v_j \rangle^2}_{= \|\text{proj}_V(x_i)\|_2^2}$$

$= \|\text{proj}_V(x_i)\|_2^2$, where $V = \text{span}(\{v_1, \dots, v_k\})$.

Digression:
A review on eigenvalues / eigenvectors.

this is important!

Let $A \in \mathbb{R}^{d \times d}$ be a $\boxed{\text{symmetric matrix}}$, i.e. $A_{ij} = A_{ji}$ $\forall i,j$.

We say $v \in \mathbb{R}^d$ is an eigenvector of A with associated eigenvalue λ if

$$Av = \lambda v.$$

Eigenvectors / eigenvalues have very nice properties

Fact: If u, v are eigenvectors of A with different eigenvalues, then $\langle u, v \rangle = 0$

Fact: (minimax theorem). Let λ_1 be the largest eigenvalue of A. Then

$$\lambda_1 = \max_{\|u\|_2 = 1} u^T A u. \longrightarrow \text{maxima achieved by eigenvector.}$$

e.g. $d = 2$, A is diagonal

$$A = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix}$$

what is the top eigenvalue of A?

$v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$

$$v^T A v = (v_1 \quad v_2) \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$= \lambda_1 v_1^2 + \lambda_2 v_2^2.$$

$$\max_{\|v\|_2 = 1} v^T A v \longrightarrow \begin{cases} \text{if } \lambda_1 > \lambda_2, = \lambda_1 \\ \text{else } \lambda_2 \end{cases}$$

what achieves this?

$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

If $\lambda_1 = \lambda_2$, anything works!

What if A is not diagonal? $\longrightarrow$ there is some basis where A is diagonal.

e.g. $$A = \begin{pmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

$\nearrow \; U \qquad\qquad \nearrow \; U^T$

rotation $\qquad\qquad$ inverse rotatio

$u_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \qquad u_2 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$. Then u_1, u_2 form orthonorma basis.

$$U^T = \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} \begin{pmatrix} u_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} u_1^T \\ u_2^T \end{pmatrix} u_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

U^T is a change of basis from $\{u_1, u_2\}$ to $\{0, 1\}$.

$$U e_1 = \begin{pmatrix} u_1 & u_2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} u_1 \end{pmatrix} \qquad U e_2 = u_2.$$

U sends $\{e_1, e_2\} \to \{u_1, u_2\}$

$$\begin{aligned} A \cdot u_1 &= U \cdot D \cdot U^T u_1 \\ &= U \cdot D \cdot e_1 \\ &= 2 \cdot U \cdot e_1 \\ &= 2 \cdot u_1 \end{aligned}$$

$\Rightarrow$ u_1 is eigenvector of A w/ eigenvalue 2

Similarly, u_2 is eig w/ eigenvalue 1.

$\hookrightarrow$ to find eigenvector/eigenvalues, find rotation that makes A diagonal.

The cols of rotation are eigenvectors, the diagonal entries are eigenvalues!

<u>Theorem</u> (Spectral theorem) For any symmetric matrix A, there are U, D s.t. $A = Q D Q^T$, where

1. D is diagonal
2. Q is an orthogonal matrix $\Rightarrow Q = \begin{pmatrix} q_1 & \cdots & q_d \end{pmatrix}$, $q_1, \dots, q_d$ orthonormal.

$D = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} \Rightarrow \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ are eigenvalues, $q_1, \dots, q_d$ are eigenvectors.

The spectral decomposition should explain this minimax principle:

$$\max_{\|u\|_2=1} u^T A u = \max_{\|u\|_2=1} u^T U D U^T u$$

might as well align u w/ largest eigenvalue.

It also generalizes to the top k eigenvalues.

$$\max_{\substack{u_1,\dots,u_k \\ \text{orthonormal}}} \sum_{j=1}^{k} u_j^T A u_j$$ $\longrightarrow$ best sol'n is to take $u_1,\dots,u_k$ being the top k eigenvectors.

unique unless eigenvalues are shared.

Connection to PCA.

Let $$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^{n \times d}$$

For any vector $v \in \mathbb{R}^d$, observe:

$$Xv = \begin{pmatrix} \langle x_1, v\rangle \\ \vdots \\ \langle x_n, v\rangle \end{pmatrix}$$

so $$\|Xv\|_2^2 = \sum_{i=1}^{n} \langle x_i, v\rangle^2.$$

so PCA objective is

$$\operatorname*{argmax}_{\substack{v_1,\dots,v_k \\ \text{orthonormal}}} \sum_{j=1}^{k} \|X v_j\|_2^2$$

$$\|Xv\|_2^2 = v^T \underbrace{X^T X}_{A} v$$

"Gram matrix"

$A \in \mathbb{R}^{d \times d}$, symmetric

$A_{ij} = \langle x_i, x_j \rangle$

then PCA objective is

$$\operatorname*{argmax}_{\substack{v_1,\dots,v_k \\ \text{orthonormal}}} \sum_{j=1}^{k} v^T A v$$ $\longrightarrow$ top k eigenvectors of A!

also explains:
- nested-ness
- some times non-unique.

Algorithms for computing eigenvectors + eigenvalues.

1). Naive method: eigenvalues are roots of

$$p(\lambda) = \det(A - \lambda I).$$

find these roots to find eigenvalues. After, to find eigenvectors, solve

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

There are more complicated "exact" methods that work in time $O(d^3)$ or sometimes $O(d^\omega)$, $\omega \leq 2.376$

Power method:

Choose $u_0 \in \mathbb{R}^d$ at random w/ $\|u_0\|_2 = 1$.
For $i = 1, \cdots, t$
let $u_i = A^i u_0$.
if $\frac{u_i}{\|u_i\|_2} \approx \frac{u_{i-1}}{\|u_{i-1}\|_2}$, terminate.
return $\frac{u_t}{\|u_t\|}$

Claim: Suppose $\lambda_1 > \lambda_2$. Then, with high probability,

$$\left\langle \frac{u_t}{\|u_t\|_2}, v_1 \right\rangle \geq 1 - O\left(\sqrt{d} \cdot \left(\frac{\lambda_2}{\lambda_1}\right)^t\right)$$

so to get ε-close, need to set $t = \frac{\log d + \log 1/\varepsilon}{\log(\lambda_2/\lambda_1)}$

pf:

$$u_0 = \underbrace{\alpha_1}_{\approx \pm \frac{1}{\sqrt{d}}} v_1 + \cdots + \underbrace{\alpha_d}_{\approx \pm 1/\sqrt{d}} v_d.$$

$$A u_0 = \alpha_1 \lambda_1 v_1 + \cdots + \alpha_d \lambda_d v_d$$

$\vdots$

$$A^t u_0 = \alpha_1 \lambda_1^t v_1 + \underbrace{\cdots + \alpha_d \lambda_d^t v_d}_{\text{growing much slower than } \lambda_1^d}.$$

$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_d$.

$$\rightarrow \frac{\lambda_1^t}{\lambda_i^t} \geq \left(\frac{\lambda_1}{\lambda_2}\right)^t \geq \frac{d^{100}}{\varepsilon}.$$

so $A^t u_0$ looks like

$$\lambda_1^t \left(\underbrace{\alpha_1 v_1}_{\text{only term that survives.}} \pm \frac{\varepsilon}{d^{100}} \alpha_2 v_2 \pm \frac{\varepsilon}{d^{100}} \alpha_2 \cdots \right)$$

What about general k?

- Can do simultaneous power method "Krylov iteration"
- Another simple algo:

  iteratively find $u_1, \dots, u_k$

  given $u_1, \dots, u_i$; to find u_{i+1}:

  do power method, but project off component

  in $u_1, \dots, u_i$.

What happens if $\lambda_1 = \lambda_2$? then any vector in $\text{span}\{v_1, v_2\}$ is eigenvector.

$$u_t = \lambda_1^t \left(\alpha_1 v_1 + \alpha_2 v_2 + \text{small} \right)$$

so you'll output something close to $\underbrace{\alpha_1 v_1 + \alpha_2 v_2}_{\text{still an eigenvector!}}$

In general, to recover v_1, will need gap between λ_1, λ_2

you can also ask for "energy" approximation, i.e.:

output any v s.t. $\|v\|_2 = 1$ and

$$v^T A v \geq (1-\varepsilon)\lambda_1 \longrightarrow t = O\left(\frac{\log d}{\varepsilon}\right)$$

for k vectors: (this implies getting an $(1-\varepsilon)$-approximation to the PCA objective)

find $v_1, \dots, v_k$ orthonormal so that

$$\sum_{i=1}^{k} v^T A v \geq (1-\varepsilon)\left(\sum_{i=1}^{k} \lambda_i\right) \longrightarrow t = \left(\frac{k \log d}{\varepsilon}\right).$$

these guarantees do not require gap