STA 221: LECTURE 3

KRISHNA BALASUBRAMANIAN

(UNIVERSITY OF CALIFORNIA, DAVIS)

- For a m by m matrix A, if

$$A\mathbf{y} = \lambda\mathbf{y},$$

  then we say

  λ is an eigenvalue of A

  $\mathbf{y}$ is the corresponding eigenvector

- Eigen decomposition defined only for square matrices.
- Eigenvalues in general can be real or complex numbers.

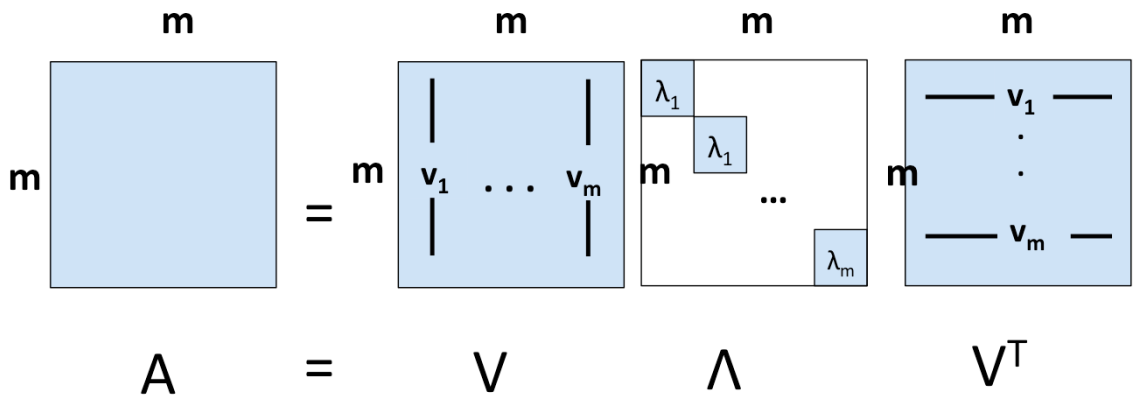
- Consider $A \in \mathbb{R}^{m \times m}$ to be a square, symmetric matrix. The eigenvalue decomposition of A is:

$$A = V \Lambda V^T, \quad V^T V = I (V \text{ is unitary}), \quad \Lambda \text{ is diagonal}$$

- $A = V \Lambda V^T \Rightarrow AV = V\Lambda$
  $\Rightarrow A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad \forall i = 1, \cdots, m$
- Each $\boldsymbol{v}_i$ is an eigenvector, and each λ_i is an eigenvalue
- Usually, we assume the diagonal numbers are organized in descending order:

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_m),$$
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$$

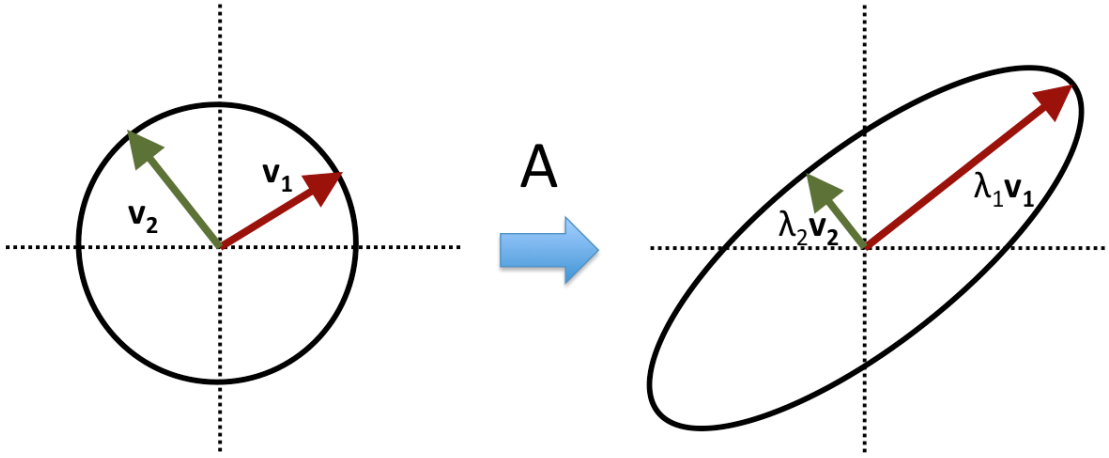
- Eigenvalue decomposition is unique when there are m unique eigenvalues.



- Each eigenvector $\boldsymbol{v}_i$ will be mapped to $A\boldsymbol{v}_i = \lambda \boldsymbol{v}_i$ after the linear transform:

  Scaling without changing the direction of eigenvectors
- $A\boldsymbol{x} = \sum_{i=1}^{m} \lambda_i \boldsymbol{v}_i (\boldsymbol{v}_i^T \boldsymbol{x})$

  Project $\boldsymbol{x}$ to eigenvectors, and then scaling each vector



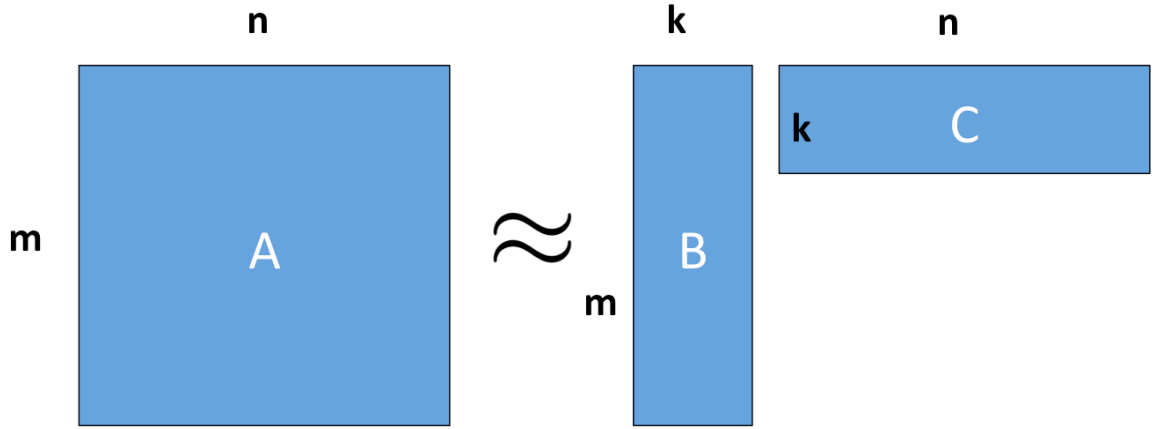
Visualization of matrix as transformation.

- ▷ Big Matrix A: Want to approximate by "simple" matrix $\widehat{A}$
- ▷ Goodness of approximation of $\widehat{A}$ can be measured by

$$\|A - \widehat{A}\|$$

(e.g., using $\|\cdot\|_F$)

- ▷ Low-rank approximation: $A \approx BC$



- Given A and k, what is the best rank-k approximation?

$$\min_{B,C \text{ are rank } k} \|A - BC\|_F$$

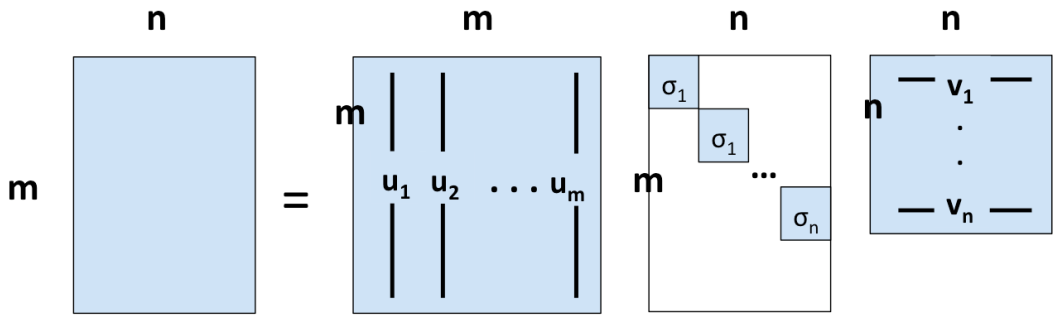
- Minimizing B and C are obtained by SVD (Singular Value Decomposition) of A

- Any real matrix $A \in \mathbb{R}^{m \times n}$ has the following Singular Value Decomposition (SVD):

$$A = U \, \Sigma \, V^T$$

  - U is a $m \times m$ unitary matrix ($U^T U = I$), columns of U are orthogonal
  - V is a $n \times n$ unitary matrix ($V^T V = I$), columns of V are orthogonal
  - Σ is a diagonal $m \times n$ matrix with **non-negative** real numbers on the diagonal.
  - Usually, we assume the diagonal numbers are organized in descending order:

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n),$$
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$$



- $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m$: left singular vectors, basis of column space

$$\boldsymbol{u}_i^T \boldsymbol{u}_j = 0 \quad \forall i \neq j, \qquad \boldsymbol{u}_i^T \boldsymbol{u}_i = 1 \quad \forall i = 1, \cdots, m$$

- $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$: right singular vectors, basis of row space

$$\boldsymbol{v}_i^T \boldsymbol{v}_j = 0 \quad \forall i \neq j, \qquad \boldsymbol{v}_i^T \boldsymbol{v}_i = 1 \quad \forall i = 1, \cdots, n$$

- SVD is unique (up to permutations, rotations)

Computing Eigenvectors

Algorithm 1 Power Method

Input: Unit-vector $\mathbf{c}^{(0)} \in \mathbb{R}^d$ (Randomly generated) and number of iterations T.

for $t = 1, \ldots, T$ **do**

$\mathbf{c}^{(t)} = \mathbf{A}\mathbf{c}^{(t-1)}$

$\mathbf{c}^{(t)} = \frac{\mathbf{c}^{(t)}}{\|\mathbf{c}^{(t)}\|_2}$ (Normalization step)

end for

- Note that, if we ignore the normalization step, we have $\mathbf{c}^{(t)} = \mathbf{A}^t \mathbf{c}^{(0)}$.
- Note that any vector could be written as a linear combination of the d eigenvectors $\boldsymbol{v}_i$.
- Specifically, the random vector $\mathbf{c}^{(0)}$ given as input to Algorithm 1 could be written as follows, for some constants $\gamma_1, \ldots, \gamma_d$.

$$\mathbf{c}^{(0)} = \gamma_1 \boldsymbol{v}_1 + \gamma_2 \boldsymbol{v}_2 + \ldots + \gamma_d \boldsymbol{v}_d$$

- Now, look at the iterates of Algorithm 1, without normalization step. We then have

$$\begin{aligned}\mathbf{c}^{(1)} = \mathbf{A}\mathbf{c}^{(0)} &= \gamma_1 \mathbf{A}\boldsymbol{v}_1 + \gamma_2 \mathbf{A}\boldsymbol{v}_2 + \ldots + \gamma_d \mathbf{A}\boldsymbol{v}_d \\ &= \gamma_1 \lambda_1 \boldsymbol{v}_1 + \gamma_2 \lambda_2 \boldsymbol{v}_2 + \ldots + \gamma_d \lambda_d \boldsymbol{v}_d.\end{aligned}$$

- Similarly, we have

$$\begin{aligned}\mathbf{c}^{(2)} = \mathbf{A}\mathbf{c}^{(1)} &= \gamma_1 \lambda_1 \mathbf{A}\boldsymbol{v}_1 + \gamma_2 \lambda_2 \mathbf{A}\boldsymbol{v}_2 + \ldots + \gamma_d \lambda_d \mathbf{A}\boldsymbol{v}_d \\ &= \gamma_1 \lambda_1^2 \boldsymbol{v}_1 + \gamma_2 \lambda_2^2 \boldsymbol{v}_2 + \ldots + \gamma_d \lambda_d^2 \boldsymbol{v}_d.\end{aligned}$$

- For the general case, we have

$$\mathbf{c}^{(t+1)} = \mathbf{A}\mathbf{c}^{(t)} = \gamma_1 \lambda_1^{(t+1)} \boldsymbol{v}_1 + \gamma_2 \lambda_2^{(t+1)} \boldsymbol{v}_2 + \ldots + \gamma_d \lambda_d^{(t+1)} \boldsymbol{v}_d.$$

Power Method

- ▷ Recall that, we have assumed λ_1 is larger than other other eigenvalues. So, when it is raised to the power $(t+1)$, the vector $\mathbf{c}^{(t+1)}$ is most correlated (or aligned) in the direction of $\boldsymbol{v}_1$.
- ▷ But if λ_2 is very close to λ_1, then t might have to be really large to get $\mathbf{c}^{(t+1)}$ correlated with $\boldsymbol{v}_1$.

Algorithm 2 Truncated Power Method

Input: Unit-vector $\mathbf{c}^{(0)} \in \mathbb{R}^d$ (Randomly generated) and number of iterations T.

for $t = 1, \ldots, T$ **do**

$\mathbf{c}^{(t)} = \mathbf{A}\mathbf{c}^{(t-1)}$

$\mathbf{c}^{(t)} = \frac{\mathbf{c}^{(t)}}{\|\mathbf{c}^{(t)}\|_2}$ (Normalization step)

Let $F^{(t)} \subset \{1, \ldots, d\}$ be the set of co-ordinates of $\mathbf{c}^{(t)}$ corresponding to top k absolute values.

Keep the values of $\mathbf{c}^{(t)}$ in the co-ordinates index by the set $F^{(t)}$ as such and set the rest of the coordinates to zero and re-normalize.

end for

In linear PCA, we try to represent each data sample as linear combination of some fixed basis vector.

$$X^{(i)} = \sum_{j=1}^{p} \mathbf{v}_j \beta_j^{(i)} = \mathbf{V}\beta^{(i)}$$

where $\mathbf{V} \in \mathbb{R}^{d \times p}$ is the matrix with the basis vectors $\mathbf{v}_j$ as its columns and $\beta_j^{(i)}$ denote the j^{th} coordinate of the vector $\beta^{(i)}$. Note that the basis vectors are fixed and does not change with i and the coefficients change with i. The reasoning behind this is that, you want to represent each sample as a different linear combination of the same basis vector.

Algorithm 3 Stochastic/Online Algorithm for PCA

Input: Unit-vector $\mathbf{v}^{(0)} \in \mathbb{R}^d$ (Randomly generated) and step-size β.

for $t = 1, \ldots, T$ **do**

Sample $X^{(t)}$ from the distribution P(X).

Let $\mathbf{v}^{(t)} = \mathbf{v}^{(t-1)} + \beta X^{(t)} {X^{(t)}}^{\top} \mathbf{v}^{(t-1)}$

$\mathbf{v}^{(t)} = \frac{\mathbf{v}^{(t)}}{\|\mathbf{v}^{(t)}\|_2}$

end for