Lecture 4

More on PCA, and the singular value decomposition

COURSE MATH 637: Mathematical Techniques in Data Science

WEBSITE https://jacobcalvert.com/teaching/math637fall2026/

INSTRUCTOR Jacob Calvert (calvert@udel.edu)

OFFICE HOURS MW 2:45pm–3:45pm, Ewing Hall 509

Announcements & Reminders

- Friday: first quiz, then problem solving and programming
- I'll hold office hours on Wed/Fri of next week due to Labor Day

Plan for today

• The linear dimension reduction problem	10 min
• Why PCA solves it	15 min
• Singular value decomposition (SVD)	25 min
• PCA via SVD	5 min

Recall: General dimension reduction problem

Problem (Dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\underset{f \in \mathcal{F},\, g \in \mathcal{G}}{\arg\min} \sum_{i=1}^{n} \| x_i - g(f(x_i)) \|^2,$$

where $\mathcal{F}$ and $\mathcal{G}$ are classes of maps from $\mathbb{R}^p \to \mathbb{R}^k$ and $\mathbb{R}^k \to \mathbb{R}^p$, resp.

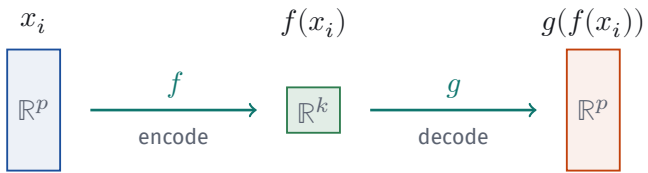
Recall: General dimension reduction problem

Problem (Dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\underset{f \in \mathcal{F},\, g \in \mathcal{G}}{\arg\min} \sum_{i=1}^{n} \| x_i - g(f(x_i)) \|^2,$$

where $\mathcal{F}$ and $\mathcal{G}$ are classes of maps from $\mathbb{R}^p \to \mathbb{R}^k$ and $\mathbb{R}^k \to \mathbb{R}^p$, resp.



Linear dimension reduction problem

Problem (Linear dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\arg\min_{E,D} \sum_{i=1}^{n} \| x_i - DE^\mathsf{T} x_i \|^2,$$

where $E, D \in \mathbb{R}^{p \times k}$ are matrices.

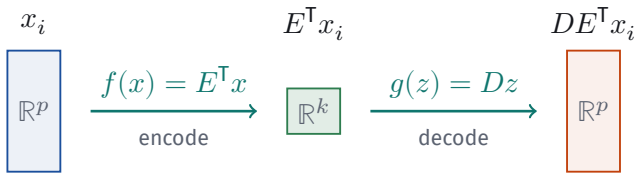
Linear dimension reduction problem

Problem (Linear dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\arg\min_{E,D} \sum_{i=1}^{n} \| x_i - DE^\mathsf{T} x_i \|^2,$$

where $E, D \in \mathbb{R}^{p \times k}$ are matrices.



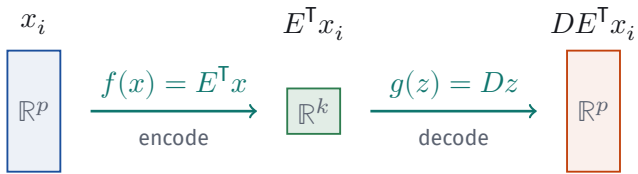
Linear dimension reduction problem

Problem (Linear dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\arg\min_{E,D} \sum_{i=1}^{n} \| x_i - DE^\mathsf{T} x_i \|^2,$$

where $E, D \in \mathbb{R}^{p \times k}$ are matrices (hence DE^T has rank $\leq k$).



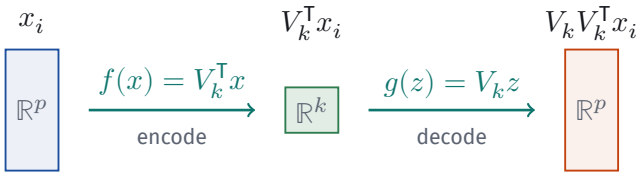
An equivalent problem

Problem (Linear dimension reduction)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$ and a dimension $k < p$, solve

$$\arg\min_{V_k} \sum_{i=1}^{n} \| x_i - V_k V_k^\mathsf{T} x_i \|^2,$$

where $V_k \in \mathbb{R}^{p \times k}$ has orthonormal columns (hence $V_k V_k^\mathsf{T}$ is a projection).

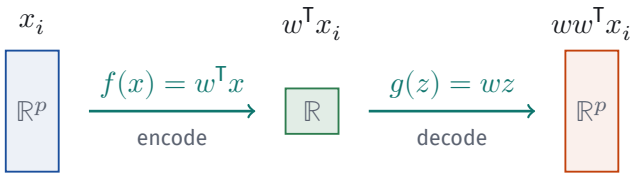


Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$, find the unit vector $w \in \mathbb{R}^p$ that minimizes

$$\sum_{i=1}^{n} \| x_i - ww^\mathsf{T} x_i \|^2.$$



Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$, find the unit vector $w \in \mathbb{R}^p$ that minimizes

$$\sum_{i=1}^{n} \| x_i - (x_i^\intercal w) w \|^2.$$

Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$, find the unit vector $w \in \mathbb{R}^p$ that minimizes

$$\sum_{i=1}^{n} \| x_i - (x_i^\mathsf{T} w) w \|^2.$$

- By Pythagorean theorem,

$$\|x_i\|^2 = (x_i^\mathsf{T} w)^2 + \|x_i - (x_i^\mathsf{T} w) w\|^2.$$

Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given examples $x_1, \dots, x_n \in \mathbb{R}^p$, find the unit vector $w \in \mathbb{R}^p$ that minimizes

$$\sum_{i=1}^{n} \| x_i - (x_i^\mathsf{T} w) w \|^2.$$

- By Pythagorean theorem,

$$\|x_i\|^2 = (x_i^\mathsf{T} w)^2 + \|x_i - (x_i^\mathsf{T} w) w\|^2.$$

- Hence, minimizing reconstruction error is the same as maximizing:

$$\sum_{i=1}^{n} (x_i^\mathsf{T} w)^2 = \sum_{i=1}^{n} (Xw)_i^2 = (Xw)^\mathsf{T}(Xw) = w^\mathsf{T} X^\mathsf{T} X w.$$

Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given a data matrix $X \in \mathbb{R}^{n \times p}$, find the unit vector $w \in \mathbb{R}^p$ that maximizes

$$w^\mathsf{T} X^\mathsf{T} X w.$$

- By Pythagorean theorem,

$$\|x_i\|^2 = (x_i^\mathsf{T} w)^2 + \|x_i - (x_i^\mathsf{T} w)w\|^2.$$

- Hence, minimizing reconstruction error is the same as maximizing:

$$\sum_{i=1}^n (x_i^\mathsf{T} w)^2 = \sum_{i=1}^n (Xw)_i^2 = (Xw)^\mathsf{T}(Xw) = w^\mathsf{T} X^\mathsf{T} X w.$$

Special case: Reduction to one dimension

Problem (Linear reduction to 1D)

Given a data matrix $X \in \mathbb{R}^{n \times p}$, find the unit vector $w \in \mathbb{R}^p$ that maximizes

$$w^\mathsf{T} X^\mathsf{T} X w.$$

Key idea: Solving the $k = 1$ case will explain why PCA solves the problem for general k.

Plan for today

- The linear dimension reduction problem — 10 min
- Why PCA solves it — 15 min
- Singular value decomposition (SVD) — 25 min
- PCA via SVD — 5 min

Solution to the 1D case

Theorem

The maximizer of $w^\mathsf{T} X^\mathsf{T} X w$ over all unit vectors $w \in \mathbb{R}^p$ is v_1, the eigenvector of $X^\mathsf{T} X$ with greatest eigenvalue λ_1.

Solution to the 1D case

Theorem

The maximizer of $w^\mathsf{T} X^\mathsf{T} X w$ over all unit vectors $w \in \mathbb{R}^p$ is v_1, the eigenvector of $X^\mathsf{T} X$ with greatest eigenvalue λ_1.

Proof. First, $w^\mathsf{T} X^\mathsf{T} X w$ is at least λ_1 because the choice $w = v_1$ gives

$$v_1^\mathsf{T}(X^\mathsf{T} X v_1) = v_1^\mathsf{T}(\lambda_1 v_1) = \lambda_1 v_1^\mathsf{T} v_1 = \lambda_1.$$

□

Solution to the 1D case

Theorem

The maximizer of $w^\mathsf{T} X^\mathsf{T} X w$ over all unit vectors $w \in \mathbb{R}^p$ is v_1, the eigenvector of $X^\mathsf{T} X$ with greatest eigenvalue λ_1.

Proof. First, $w^\mathsf{T} X^\mathsf{T} X w$ is at least λ_1 because the choice $w = v_1$ gives

$$v_1^\mathsf{T}(X^\mathsf{T} X v_1) = v_1^\mathsf{T}(\lambda_1 v_1) = \lambda_1 v_1^\mathsf{T} v_1 = \lambda_1.$$

Second, write w in the eigenbasis of $X^\mathsf{T} X$ as $w = \sum_{i=1}^p (w^\mathsf{T} v_i) v_i$. Then,

$$w^\mathsf{T} X^\mathsf{T} X w = \sum_{i=1}^p (w^\mathsf{T} v_i) w^\mathsf{T} X^\mathsf{T} X v_i = \sum_{i=1}^p \lambda_i (w^\mathsf{T} v_i)^2 \leq \lambda_1 \underbrace{\sum_{i=1}^p (w^\mathsf{T} v_i)^2}_{= \|w\|^2 = 1} = \lambda_1.$$

□

What the proof actually says

Key idea: The quantity $w^\mathsf{T} X^\mathsf{T} X w$ is a weighted average of the eigenvalues:

$$w^\mathsf{T} X^\mathsf{T} X w = \sum_{i=1}^{p} q_i \lambda_i, \qquad q_i = (w^\mathsf{T} v_i)^2 \geq 0, \qquad \sum_{i=1}^{p} q_i = \|w\|^2 = 1,$$

where q_i says how much overlap the direction w has with v_i.

- Being an average, $w^\mathsf{T} X^\mathsf{T} X w$ is bounded by the smallest and largest eigenvalues:
$$\lambda_p \leq w^\mathsf{T} X^\mathsf{T} X w \leq \lambda_1.$$
- To make it as large as possible, put all the weight on v_1.

Linear dimension reduction to 2D

Call $w^{(1)} = v_1$. Now find:

$$w^{(2)} = \arg\max_{\substack{\|w\|=1 \\ w \perp w^{(1)}}} w^\mathsf{T} X^\mathsf{T} X w.$$

- The same argument (with minor tweaks) works!

Linear dimension reduction in general

Having found $w^{(1)}, \ldots, w^{(k)}$, find:

$$w^{(k+1)} = \underset{\substack{\|w\|=1 \\ w \perp w^{(1)}, \ldots, w^{(k)}}}{\arg\max} \; w^\mathsf{T} X^\mathsf{T} X w$$

- The same argument (with minor tweaks) works!
- So $w^{(k)} = v_k$, with variation λ_k, for all $k < p$.

Key idea: One eigendecomposition of $X^\mathsf{T}X$ does PCA of X for all $k < p$ at once, and in order!

Conclusion

Problem (Linear dimension reduction)

Given a data matrix $X \in \mathbb{R}^{n \times p}$ and a dimension $k < p$, solve

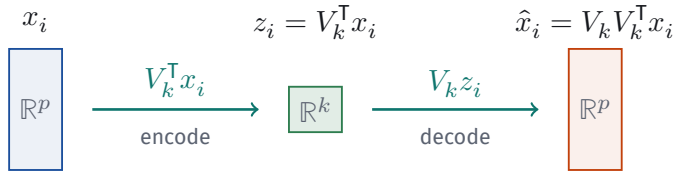
$$\arg\min_{V_k} \left\| X - X V_k V_k^{\mathsf{T}} \right\|^2,$$

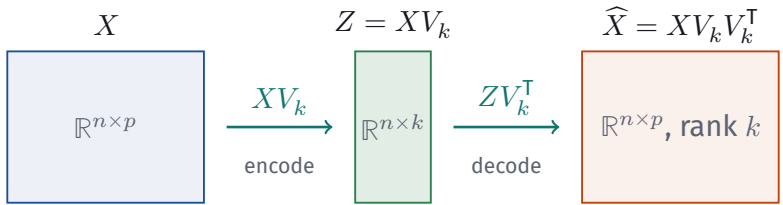
where $V_k \in \mathbb{R}^{p \times k}$ has orthonormal columns.

Theorem (PCA solves linear dimension reduction)

The solution is $V_k = [v_1 \cdots v_k]$, where $v_1, \ldots, v_k$ are the top k eigenvectors of $X^{\mathsf{T}} X$, in which case the reconstruction error equals $\sum_{i>k} \lambda_i$.

Note: The algebra looks different for the data matrix





Plan for today

- The linear dimension reduction problem — 10 min
- Why PCA solves it — 15 min
- Singular value decomposition (SVD) — 25 min
- PCA via SVD — 5 min

Enter: the singular value decomposition

Although we *derived* everything about PCA from the covariance matrix $X^\mathsf{T}X$, it is generally unwise to *compute* with it!

- Forming $X^\mathsf{T}X$ squares the "condition number," which can easily lead to a loss of numerical precision.

Enter: the singular value decomposition

Although we *derived* everything about PCA from the covariance matrix $X^\mathsf{T}X$, it is generally unwise to *compute* with it!

- Forming $X^\mathsf{T}X$ squares the "condition number," which can easily lead to a loss of numerical precision.
- Standard implementations of PCA instead use the singular value decomposition (SVD).

Enter: the singular value decomposition

Although we *derived* everything about PCA from the covariance matrix $X^\mathsf{T}X$, it is generally unwise to *compute* with it!

- Forming $X^\mathsf{T}X$ squares the "condition number," which can easily lead to a loss of numerical precision.
- Standard implementations of PCA instead use the singular value decomposition (SVD).
- SVD is *much more* than just a means for numerically implementing PCA.

Enter: the singular value decomposition

Although we *derived* everything about PCA from the covariance matrix $X^\mathsf{T}X$, it is generally unwise to *compute* with it!

- Forming $X^\mathsf{T}X$ squares the "condition number," which can easily lead to a loss of numerical precision.
- Standard implementations of PCA instead use the singular value decomposition (SVD).
- SVD is *much more* than just a means for numerically implementing PCA.

> **Key idea:** SVD generalizes the spectral theorem to *all* matrices!

Definition: SVD

Theorem (Singular value decomposition)

Every matrix $X \in \mathbb{R}^{n \times p}$ can be factored as

$$X = U \Sigma V^{\mathsf{T}},$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{p \times p}$ are orthogonal ($U^{\mathsf{T}} U = I_n$, $V^{\mathsf{T}} V = I_p$) and $\Sigma \in \mathbb{R}^{n \times p}$ is diagonal with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,p)} \geq 0$.

Definition: SVD

Theorem (Singular value decomposition)

Every matrix $X \in \mathbb{R}^{n \times p}$ can be factored as

$$X = U \Sigma V^{\mathsf{T}},$$

where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{p \times p}$ are orthogonal ($U^{\mathsf{T}}U = I_n$, $V^{\mathsf{T}}V = I_p$) and $\Sigma \in \mathbb{R}^{n \times p}$ is diagonal with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,p)} \geq 0$.

- Columns $u_1, \ldots, u_n$ of U: "left singular vectors" form basis of $\mathbb{R}^n$
- Columns $v_1, \ldots, v_p$ of V: "right singular vectors" form basis of $\mathbb{R}^p$
- Diagonal entries σ_j: "singular values"

Example: What an SVD looks like

Example 4.12 (Vectors and the SVD)
Consider a mapping of a square grid of vectors $\mathcal{X} \in \mathbb{R}^2$ that fit in a box of size 2×2 centered at the origin. Using the standard basis, we map these vectors using

$$\boldsymbol{A} = \begin{bmatrix} 1 & -0.8 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \tag{4.67a}$$

$$= \begin{bmatrix} -0.79 & 0 & -0.62 \\ 0.38 & -0.78 & -0.49 \\ -0.48 & -0.62 & 0.62 \end{bmatrix} \begin{bmatrix} 1.62 & 0 \\ 0 & 1.0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.78 & 0.62 \\ -0.62 & -0.78 \end{bmatrix}. \tag{4.67b}$$

From book by Deisenroth–Faisal–Ong

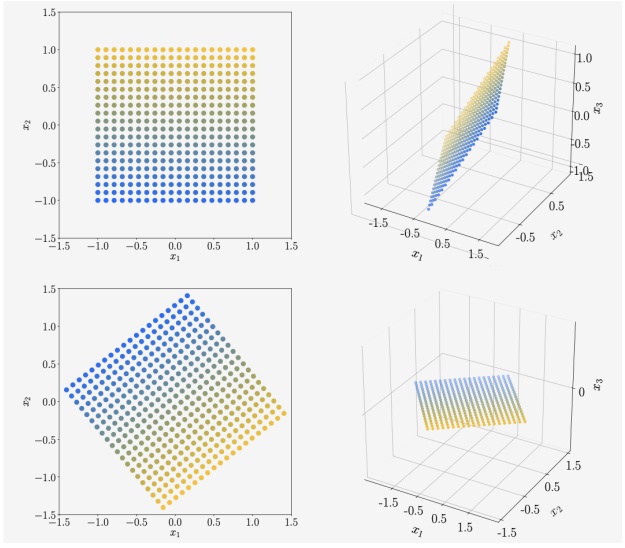


Fig. 4.9 of Deisenroth–Faisal–Ong

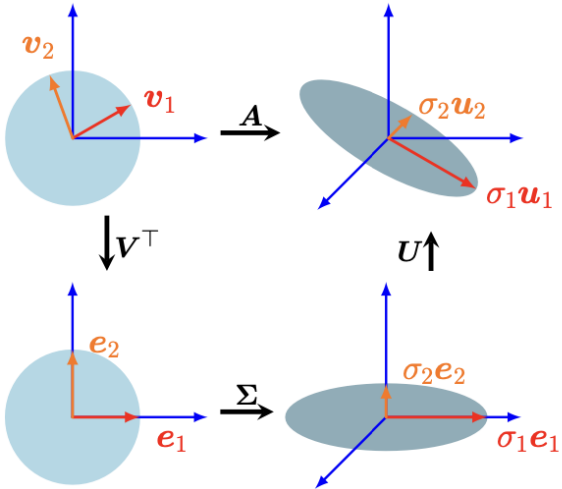


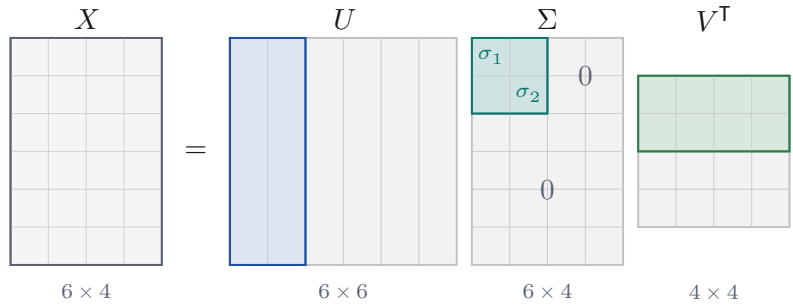
Fig. 4.8 of Deisenroth–Faisal–Ong

Every rectangular matrix: rotate, stretch, rotate

Read $X = U\Sigma V^\mathsf{T}$ acting on a vector, right to left:

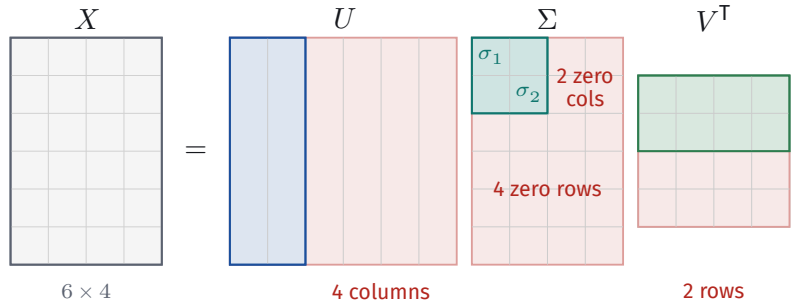
- V^T rotates: it re-expresses the input in the basis $v_1, \dots, v_p$
- Σ stretches coordinate j by σ_j (and changes the dimension)
- U rotates the result into the basis $u_1, \dots, u_n$

Σ often contains large blocks of zeros



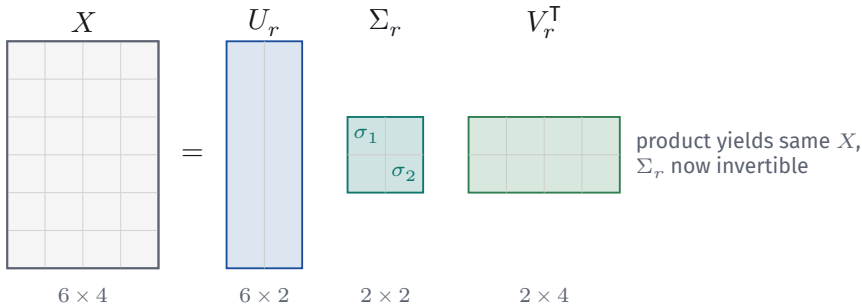
Here $n = 6$, $p = 4$, and X has rank $r = 2$: only σ_1, σ_2 are nonzero.

The corresponding parts of U, V^T don't contribute



- Column j of U is only ever multiplied by row j of Σ
- Row j of V^T is only ever multiplied by column j of Σ
- Past $j = r$ these columns/rows are zero, so they don't contribute to X

Deleting them gives the "compact" SVD



- r is the rank of X
- $U_r \in \mathbb{R}^{n \times r}$, $\Sigma_r \in \mathbb{R}^{r \times r}$, and $V_r^\mathsf{T} \in \mathbb{R}^{r \times p}$ are the pieces of the compact SVD

We can write X as a sum of outer products

The compact form of the SVD as a sum of r outer products:

$$X = U_r \Sigma_r V_r^\mathsf{T} = \sum_{j=1}^{r} \sigma_j \, u_j v_j^\mathsf{T}$$

- Each term $\sigma_j u_j v_j^\mathsf{T}$ is an $n \times p$ matrix of rank *one*
- They are listed in order of importance: $\sigma_1 \geq \sigma_2 \geq \cdots$

We can write X as a sum of outer products

The compact form of the SVD as a sum of r outer products:

$$X = U_r \Sigma_r V_r^\mathsf{T} = \sum_{j=1}^{r} \sigma_j \, u_j v_j^\mathsf{T}$$

- Each term $\sigma_j u_j v_j^\mathsf{T}$ is an $n \times p$ matrix of rank *one*
- They are listed in order of importance: $\sigma_1 \geq \sigma_2 \geq \cdots$

Coming up: Keeping only the first $k < r$ terms is the truncated SVD. It is the optimal rank-k approximation to X!

Compare: Two rank-one expansions

Spectral theorem

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

$$A = V \Lambda V^\mathsf{T} = \sum_{j=1}^{p} \lambda_j \, v_j v_j^\mathsf{T}$$

- *one* orthonormal basis
- λ_j may be negative

Singular value decomp.

For any $X \in \mathbb{R}^{n \times p}$,

$$X = U \Sigma V^\mathsf{T} = \sum_{j=1}^{r} \sigma_j \, u_j v_j^\mathsf{T}$$

- *two* orthonormal bases
- $\sigma_j \geq 0$ always

Example: Movie ratings

	Matrix	Alien	Star Wars	Casablanca	Titanic
Ana	1	1	1	0	0
Ben	3	3	3	0	0
Cai	4	4	4	0	0
Dee	5	5	5	0	0
Eve	0	2	0	4	4
Fay	0	0	0	5	5
Gus	0	1	0	2	2

Data matrix $X \in \mathbb{R}^{7\times 5}$ has one row per person containing their ratings. Ratings on a scale of 1–5, and 0 means "did not rate."

Example: SVD finds "genres"

The singular values are $\sigma_1 = 12.5$, $\sigma_2 = 9.5$, $\sigma_3 = 1.3$, $\sigma_4 = \sigma_5 = 0$, and the first three right singular vectors are

	Matrix	Alien	Star Wars	Casablanca	Titanic
v_1	0.56	0.59	0.56	0.09	0.09
v_2	−0.13	0.03	−0.13	0.70	0.70
v_3	−0.41	0.80	−0.41	−0.09	−0.09

- v_1 focuses on the three films: call it sci-fi concept

Example: SVD finds “genres”

The singular values are $\sigma_1 = 12.5$, $\sigma_2 = 9.5$, $\sigma_3 = 1.3$, $\sigma_4 = \sigma_5 = 0$, and the first three right singular vectors are

	Matrix	Alien	Star Wars	Casablanca	Titanic
v_1	0.56	0.59	0.56	0.09	0.09
v_2	−0.13	0.03	−0.13	0.70	0.70
v_3	−0.41	0.80	−0.41	−0.09	−0.09

- v_1 focuses on the three films: call it sci-fi concept
- v_2 focuses on the last two: the romance concept

Example: SVD finds "genres"

The singular values are $\sigma_1 = 12.5$, $\sigma_2 = 9.5$, $\sigma_3 = 1.3$, $\sigma_4 = \sigma_5 = 0$, and the first three right singular vectors are

	Matrix	Alien	Star Wars	Casablanca	Titanic
v_1	0.56	0.59	0.56	0.09	0.09
v_2	−0.13	0.03	−0.13	0.70	0.70
v_3	−0.41	0.80	−0.41	−0.09	−0.09

- v_1 focuses on the three films: call it sci-fi concept
- v_2 focuses on the last two: the romance concept
- v_3 is Alien against the other two (horror?), but σ_3 is tiny

Example: who likes which concept

$$X \approx \underbrace{\sigma_1 u_1 v_1^\mathsf{T}}_{\text{sci-fi}} + \underbrace{\sigma_2 u_2 v_2^\mathsf{T}}_{\text{romance}}$$

	Ana	Ben	Cai	Dee	Eve	Fay	Gus
u_1	0.14	0.41	0.55	0.69	0.15	0.07	0.08
u_2	−0.02	−0.07	−0.09	−0.12	0.59	0.73	0.30

- v_j says how much of concept j is in each movie
- u_j says how much each person likes concept j
- σ_j says how strong concept j is in the data overall

Example: two concepts, all 35 ratings

Keeping only the two strongest concepts, $\widehat{X} = \sigma_1 u_1 v_1^\mathsf{T} + \sigma_2 u_2 v_2^\mathsf{T}$:

	Matrix	Alien	Star Wars	Casablanca	Titanic
Ana	1.0	1.0	1.0	0.0	0.0
Ben	3.0	3.0	3.0	0.0	0.0
Cai	4.0	4.0	4.0	0.0	0.0
Dee	5.0	5.1	5.0	0.0	0.0
Eve	0.4	1.3	0.4	4.1	4.1
Fay	−0.4	0.7	−0.4	4.9	4.9
Gus	0.2	0.6	0.2	2.0	2.0

Two numbers per person and two per movie reproduce all 35 ratings, to within $\left\|X - \widehat{X}\right\|_F^2 = \sigma_3^2 = 1.8$ out of $\sum_j \sigma_j^2 = 248$.

Note: From reconstruction to recommendation

In the literature: The Netflix Prize (2006–2009) offered $1M for a 10% improvement over Netflix's own recommender. The winning entries were built on exactly this idea: approximate the ratings matrix by a low-rank one.

Note: From reconstruction to recommendation

In the literature: The Netflix Prize (2006–2009) offered $1M for a 10% improvement over Netflix's own recommender. The winning entries were built on exactly this idea: approximate the ratings matrix by a low-rank one.

Watch out: We just treated "did not rate" as a rating of 0. Real systems instead *fit* U, Σ, V to the observed entries only, and read off the rest.

Note: From reconstruction to recommendation

In the literature: The Netflix Prize (2006–2009) offered $1M for a 10% improvement over Netflix's own recommender. The winning entries were built on exactly this idea: approximate the ratings matrix by a low-rank one.

Watch out: We just treated "did not rate" as a rating of 0. Real systems instead *fit* U, Σ, V to the observed entries only, and read off the rest.

Implementing a recommendation system for "real" data would make for a fun project!

Plan for today

- The linear dimension reduction problem — 10 min
- Why PCA solves it — 15 min
- Singular value decomposition (SVD) — 25 min
- PCA via SVD — 5 min

PCA via the SVD

Let $X \in \mathbb{R}^{n \times p}$ be centered with compact SVD $X = U_r \Sigma_r V_r^\mathsf{T}$.

	from $X^\mathsf{T}X$	from the SVD of X
loadings	eigenvectors $v_1, \ldots, v_k$ of $X^\mathsf{T}X$	right singular vectors, V_k
variation of PC$_j$	eigenvalue λ_j	σ_j^2
scores	$Z = XV_k$	$Z = U_k\Sigma_k$
PVE(k)	$\sum_{j \le k} \lambda_j / \sum_j \lambda_j$	$\sum_{j \le k} \sigma_j^2 / \sum_j \sigma_j^2$

Algorithm: PCA, as it is actually computed

Input. Data matrix $X \in \mathbb{R}^{n \times p}$, a dimension $k \in \{1, \dots, p-1\}$.

Output. Scores $Z \in \mathbb{R}^{n \times k}$, loadings $V_k \in \mathbb{R}^{p \times k}$.

1. Center: $X \leftarrow X - \frac{1}{n}\mathbf{1}\mathbf{1}^\mathsf{T}X$
2. Factor: compute the SVD $X = U\Sigma V^\mathsf{T}$ (never form $X^\mathsf{T}X$)
3. Truncate: $V_k = [\, v_1 \ \cdots \ v_k \,]$, $U_k = [\, u_1 \ \cdots \ u_k \,]$
4. Project: $Z = XV_k = U_k\Sigma_k$

> **Note:** The scikit-learn package in Python implements PCA by applying `np.linalg.svd` to the centered data matrix.