COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2026.

Lecture 15

Logistics

- Problem Set 2 grades have been released.
- Typical online quiz and in class quiz this week.
- Practice exams for Midterm 2 posted on Canvas. Study guide will be posted soon.

Summary

Last Class:

- Finish up proof of the JL Lemma.
- Start on motivation of vector quantization and application of JL to this problem.

(distributional JL
$\mathbb{E}\|\Pi y\|_2^2 = \|y\|_2^2$)

This Class:

- Finish up JL for vector quantization.
- Start on data dependent dimensionality reduction (first lecture in series on PCA/SVD/optimal low-rank approximation)

Applications of JL to Vector Quantization

Vector Quantization Problem

Have many stored vectors $\vec{k}_1, \ldots, \vec{k}_n \in \mathbb{R}^d$ and receive queries $\vec{q} \in \mathbb{R}^d$.

Goal: Estimate $\langle \vec{q}, \vec{k}_i \rangle$ for each stored vector while minimizing space usage/query time (similar ideas could apply e.g., to estimating $\|\vec{q} - \vec{k}_i\|_2$.)

Vector quantization: Represent each $\vec{k}_i$ with a compressed version, at the cost of only giving an approximate inner product.

Vector Quantization Problem

Have many stored vectors $\vec{k}_1, \ldots, \vec{k}_n \in \mathbb{R}^d$ and receive queries $\vec{q} \in \mathbb{R}^d$.

Goal: Estimate $\langle \vec{q}, \vec{k}_i \rangle$ for each stored vector while minimizing space usage/query time (similar ideas could apply e.g., to estimating $\|\vec{q} - \vec{k}_i\|_2$.)

Vector quantization: Represent each $\vec{k}_i$ with a compressed version, at the cost of only giving an approximate inner product.

Applications:

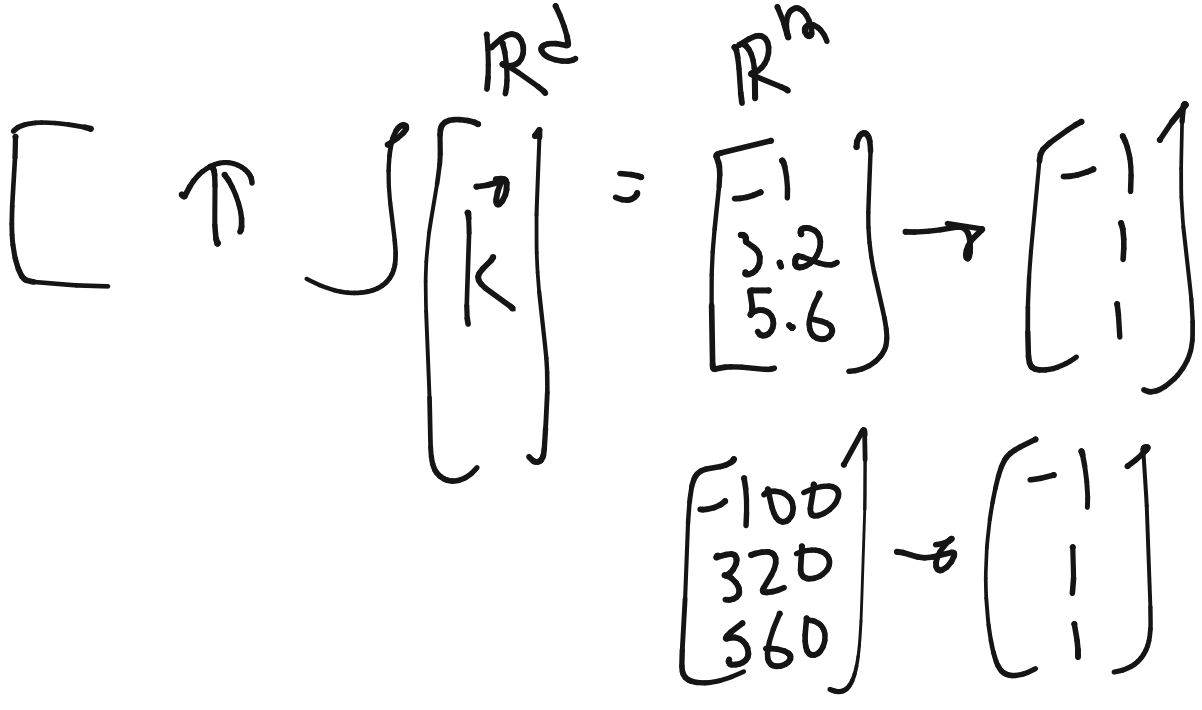
- Compressing storage for nearest neighbor search algorithms.
- Key-value cache compression for computing attention in large language models.
- Here, $\vec{q}$ is a 'query' embedding for the next token, and $\vec{k}_1, \ldots, \vec{k}_n$ are 'key' embeddings for all previous tokens in the context window.

Quantized JL for Vector Quantization

Quantized JL

A JL sketch $\mathbf{\Pi}\vec{k}$ stores m real numbers. The idea of quantized JL (QJL) [Zandieh, Daliri, Han '24] is to just keep the signs of these numbers.

$$\vec{k} \in \mathbb{R}^d \to \mathbf{\Pi}\vec{k} \in \mathbb{R}^m \to \mathsf{sign}(\mathbf{\Pi}\vec{k}) \in \{\pm 1\}^m.$$



Quantized JL

A JL sketch $\mathbf{\Pi}\vec{k}$ stores m real numbers. The idea of quantized JL (QJL) [Zandieh, Daliri, Han '24] is to just keep the signs of these numbers.

$$\vec{k} \in \mathbb{R}^d \to \mathbf{\Pi}\vec{k} \in \mathbb{R}^m \to \mathbf{sign}(\mathbf{\Pi}\vec{k}) \in \{\pm 1\}^m.$$

- Will also store the norm $\|\vec{k}\|_2$ to preserve magnitude information (one real number).

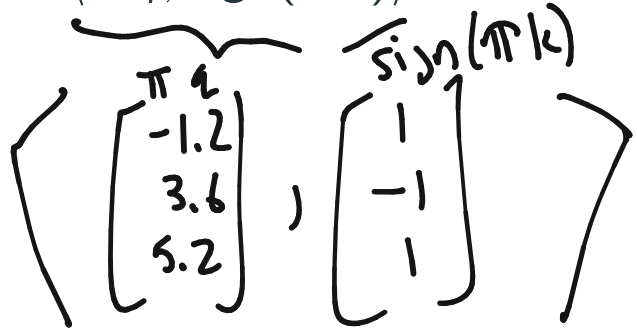
Quantized JL

A JL sketch $\mathbf{\Pi}\vec{k}$ stores m real numbers. The idea of quantized JL (QJL) [Zandieh, Daliri, Han '24] is to just keep the signs of these numbers.

$$\vec{k} \in \mathbb{R}^d \to \mathbf{\Pi}\vec{k} \in \mathbb{R}^m \to \mathsf{sign}(\mathbf{\Pi}\vec{k}) \in \{\pm 1\}^m.$$

- Will also store the norm $\|\vec{k}\|_2$ to preserve magnitude information (one real number).
- Given a query $\vec{q}$, estimate

$$\langle \vec{q}, \vec{k} \rangle \approx \frac{\sqrt{\pi/2}}{m} \cdot \|\vec{k}\|_2 \cdot \langle \mathbf{\Pi}\vec{q}, \mathsf{sign}(\mathbf{\Pi}\vec{k}) \rangle.$$



Quantized JL

A JL sketch $\mathbf{\Pi}\vec{k}$ stores m real numbers. The idea of quantized JL (QJL) [Zandieh, Daliri, Han '24] is to just keep the signs of these numbers.

$$\vec{k} \in \mathbb{R}^d \to \mathbf{\Pi}\vec{k} \in \mathbb{R}^m \to \textbf{sign}(\mathbf{\Pi}\vec{k}) \in \{\pm 1\}^m.$$

- Will also store the norm $\|\vec{k}\|_2$ to preserve magnitude information (one real number).
- Given a query $\vec{q}$, estimate

$$\langle q, \vec{k}\rangle \approx \frac{\sqrt{\pi/2}}{m} \cdot \|\vec{k}\|_2 \cdot \langle \mathbf{\Pi}\vec{q}, \textbf{sign}(\mathbf{\Pi}\vec{k})\rangle.$$

- **Analysis Goal:** Prove that this is correct in expectation and has small variance/concentrates well around its expectation.

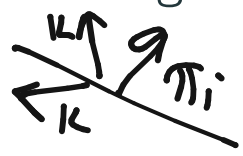
Connection to Other Methods

A variety of other vector quantization/KV Cache compression methods use similar ideas including:

- RabitQ [Gao, Long ‘24]
- TurboQuant (Google) [Zandieh, Daliri, Hadian, Mirrokni ‘25]

Connection to Other Methods

A variety of other vector quantization / KV Cache compression methods use similar ideas including:





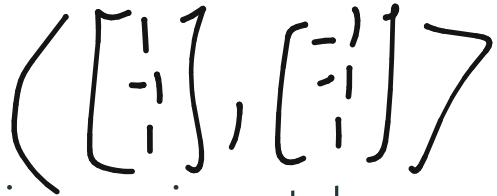
- RabitQ [Gao, Long '24]
- TurboQuant (Google) [Zandieh, Daliri, Hadian, Mirrokni '25]

Also closely related to SimHash (LSH for cosine similarity).

- $\textbf{sign}(\mathbf{\Pi}\vec{k})$ tells you exactly which side of the random hyperplanes normal to the rows of $\mathbf{\Pi}$ that $\vec{k}$ lies on. So

  $$\langle \textbf{sign}(\mathbf{\Pi}\vec{q}), \textbf{sign}(\mathbf{\Pi}\vec{k}) \rangle$$

  is exactly the number of SimHash matches minus mismatches over m trials.



$\langle \pi q, \text{sign}(\pi k) \rangle$

Connection to Other Methods

A variety of other vector quantization / KV Cache compression methods use similar ideas including:

cosine similarity = inner product

- RabitQ [Gao, Long '24]
- TurboQuant (Google) [Zandieh, Daliri, Hadian, Mirrokni '25]

Also closely related to SimHash (LSH for cosine similarity).

- $\textbf{sign}(\mathbf{\Pi}\vec{k})$ tells you exactly which side of the random hyperplanes normal to the rows of $\mathbf{\Pi}$ that $\vec{k}$ lies on. So

$$\langle \textbf{sign}(\mathbf{\Pi}\vec{q}), \textbf{sign}(\mathbf{\Pi}\vec{k}) \rangle$$

  is exactly the number of SimHash matches minus mismatches over m trials.
- In QJL, only one vector is quantized: $\langle \mathbf{\Pi}\vec{k}, \textbf{sign}(\mathbf{\Pi}\vec{q}) \rangle$. This allows expectation to directly scale with the dot product $\cos(\theta(\vec{q}, \vec{v}))$ instead of the angle $\theta(\vec{q}, \vec{v})$ between the vectors.

QJL Expectation Analysis

QJL Estimator:

$$\langle q, \vec{k} \rangle \approx \frac{\sqrt{\pi/2}}{m} \cdot \|\vec{k}\|_2 \cdot \langle \mathbf{\Pi}\vec{q}, \text{sign}(\mathbf{\Pi}\vec{k}) \rangle.$$

$$\langle \Pi q, \text{sign}(\Pi k) \rangle = \sum_{j=1}^{m} [\Pi q](j) \cdot \text{sign}(\Pi k(j))$$

$$[\Pi \quad [\Pi q] = [\langle \pi_j, q \rangle]$$

QJL Expectation Analysis

QJL Estimator:

$$\langle q, \vec{k} \rangle \approx \frac{\sqrt{\pi/2}}{m} \cdot \|\vec{k}\|_2 \cdot \langle \mathbf{\Pi}\vec{q}, \text{sign}(\mathbf{\Pi}\vec{k}) \rangle.$$

- Let $\boldsymbol{\pi}_j$ be the j^{th} row of $\mathbf{\Pi}$ and

$$\mathsf{X}_j = \sqrt{\pi/2} \cdot \|\vec{k}\|_2 \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

- Then the QJL estimator is just the average $\frac{1}{m} \sum_{j=1}^{m} \mathsf{X}_j$.

QJL Expectation Analysis

QJL Estimator:

$$\langle q, \vec{k} \rangle \approx \frac{\sqrt{\pi/2}}{m} \cdot \|\vec{k}\|_2 \cdot \langle \mathbf{\Pi}\vec{q}, \text{sign}(\mathbf{\Pi}\vec{k}) \rangle.$$

- Let $\boldsymbol{\pi}_j$ be the j^{th} row of $\mathbf{\Pi}$ and

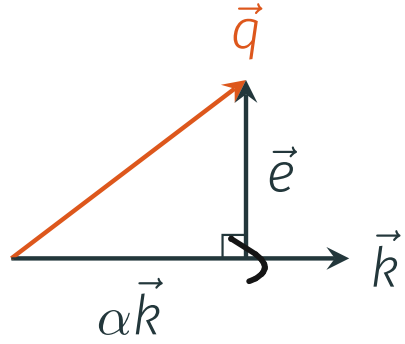
$$\mathsf{X}_j = \sqrt{\pi/2} \cdot \|\vec{k}\|_2 \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

- Then the QJL estimator is just the average $\frac{1}{m}\sum_{j=1}^{m} \mathsf{X}_j$.
- The X_j are identically distributed, so our goal is to show $\mathbb{E}[\mathsf{X}_j] = \langle \vec{q}, \vec{k} \rangle$, and later to bound $\mathsf{Var}[\mathsf{X}_j]$.
- Assume without loss of generality that $\|\vec{k}\|_2 = 1$.

QJL Expectation Analysis

Write the query as

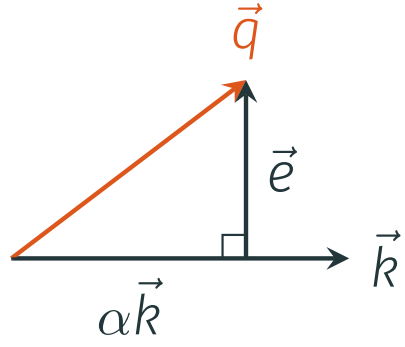
$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



QJL Expectation Analysis

Write the query as

$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



Can then use linearity to write:

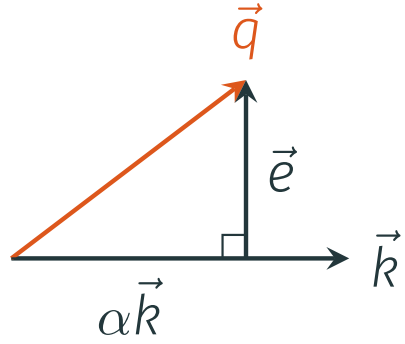
$$\langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) = \alpha \langle \boldsymbol{\pi}_j, \vec{k} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)$$

$\langle \pi_j, q \rangle = \langle \pi_j, \alpha k \rangle + \langle \pi_j, e \rangle$

QJL Expectation Analysis

Write the query as

$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



Can then use linearity to write:

$$\begin{aligned}\langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) &= \alpha \langle \boldsymbol{\pi}_j, \vec{k} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) \\ &= \alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).\end{aligned}$$

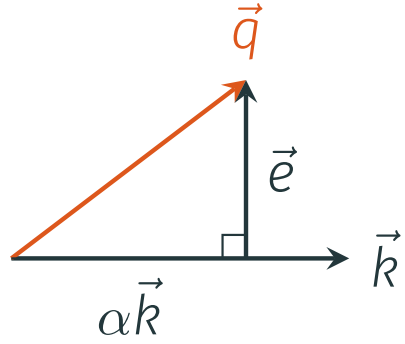
$\langle \pi_j, \vec{k} \rangle \sim N(0, \|k\|_2^2)$

$N(0,1)$

QJL Expectation Analysis

Write the query as

$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



Can then use linearity to write:

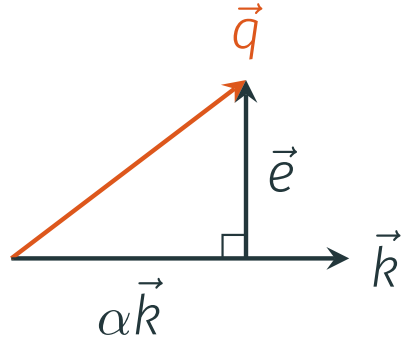
$$\begin{aligned}\langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) &= \alpha \langle \boldsymbol{\pi}_j, \vec{k} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) \\ &= \alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).\end{aligned}$$

- $\mathbb{E}[|\langle \boldsymbol{\pi}_j, \vec{k} \rangle|]$.

QJL Expectation Analysis

Write the query as

$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



Can then use linearity to write:

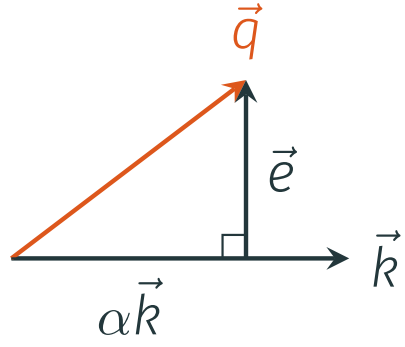
$$\begin{aligned}\langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) &= \alpha \langle \boldsymbol{\pi}_j, \vec{k} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) \\ &= \alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).\end{aligned}$$

- $\mathbb{E}[|\langle \boldsymbol{\pi}_j, \vec{k} \rangle|] = \mathbb{E}[|\mathcal{N}(0,1)|] = \sqrt{2/\pi}.$

QJL Expectation Analysis

Write the query as

$$\vec{q} = \alpha\vec{k} + \vec{e}, \qquad \alpha = \langle \vec{q}, \vec{k} \rangle, \quad \vec{e} \perp \vec{u}.$$



Can then use linearity to write:

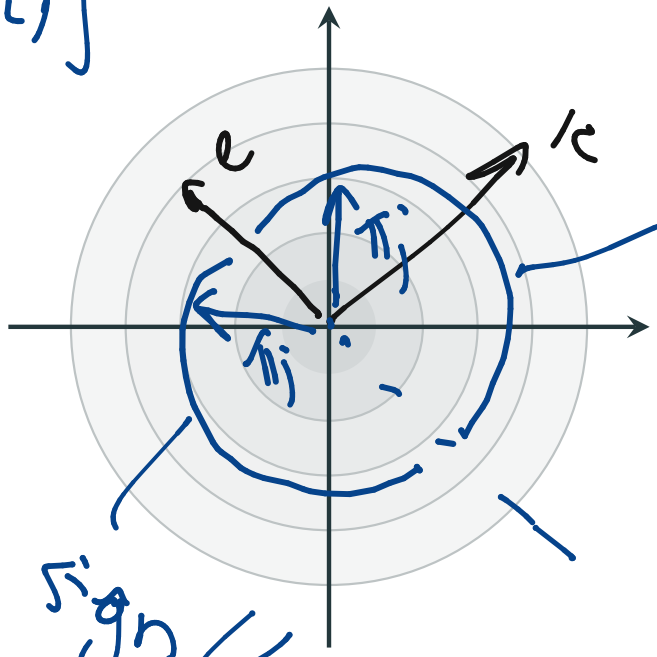
$$\begin{aligned}\langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) &= \alpha \langle \boldsymbol{\pi}_j, \vec{k} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle) \\ &= \alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).\end{aligned}$$

- $\mathbb{E}[|\langle \boldsymbol{\pi}_j, \vec{k} \rangle|] = \mathbb{E}[|\mathcal{N}(0,1)|] = \sqrt{2/\pi}$.
- $\mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = 0$.

Independence of Gaussian Projections

$\mathbb{E}[\langle \pi_j, \vec{e} \rangle \cdot \text{sign}(\langle \pi_j, \vec{k} \rangle)] = 0$. Why?

Gaussian density on $\mathbb{R}^2$:



$\mathbb{E}[\langle \pi_j, e \rangle \mid \text{sign}(\langle \pi_j, k \rangle)]$

$= \mathbb{E}[\langle \pi_j, e \rangle]$

$\text{sign}(\langle \pi_j, k \rangle) = 1$

$\mathbb{E}[\langle \pi e, \pi v \rangle] = 0$

$\text{sign}(\langle \pi_j, k \rangle) = -1$

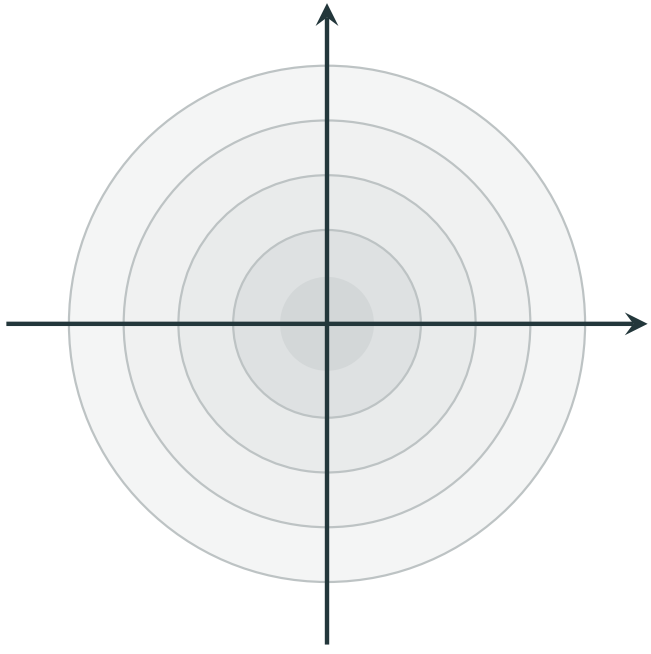
Not:

$\langle \pi_j, e \rangle \cdot \langle \pi_j, k \rangle \neq 0$

Independence of Gaussian Projections

$\mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = 0$. Why?

Gaussian density on $\mathbb{R}^2$:

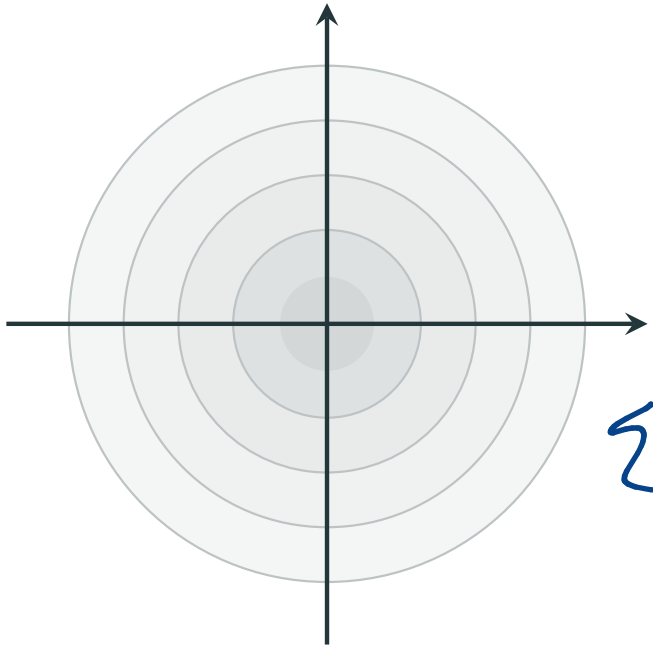


So $\langle \boldsymbol{\pi}_j, \vec{e} \rangle$ and $\mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)$ are independent.

Independence of Gaussian Projections

$\mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = 0$. Why?

Gaussian density on $\mathbb{R}^2$:



$\mathbb{E}[\langle \pi_j, e \rangle]$

$= \sum \Pr(\pi_j = \pi) \cdot \langle \pi, e \rangle$

$\Pr(\pi_j = \pi) = \Pr(\pi_j = -\pi)$

$\sum \Pr(\pi_j = \pi) \cdot [\langle \pi, e \rangle + \langle -\pi, e \rangle]$

$= 0$

So $\langle \boldsymbol{\pi}_j, \vec{e} \rangle$ and $\mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)$ are independent. Gives:

$$\mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{e} \rangle] \cdot \mathbb{E}[\mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = 0 \cdot 0 = 0.$$

QJL Expectation Analysis

$$\mathbb{E}[\sqrt{\pi/2} \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)]$$

$$\sqrt{\pi/2} \cdot \left(\alpha \cdot \sqrt{2/\pi} \cdot +0 \right)$$

$$= \alpha = \langle \vec{q}, \vec{k} \rangle.$$

QJL Expectation Analysis

$$\mathbb{E}[\sqrt{\pi/2} \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)]$$

$$\sqrt{\pi/2} \cdot \left(\alpha \cdot \sqrt{2/\pi} \cdot +0 \right)$$

$$= \alpha = \langle \vec{q}, \vec{k} \rangle.$$

Variance analysis is also straightforward. Recall that:

$$\mathsf{X}_j = \sqrt{\pi/2}\, \langle \boldsymbol{\pi}_j, \vec{q} \rangle \, \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

Squaring removes the sign, so

$$\mathbb{E}[\mathsf{X}_j^2] = \frac{\pi}{2} \cdot \mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{q} \rangle^2]$$

QJL Expectation Analysis

$$\mathbb{E}[\sqrt{\pi/2} \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)]$$

$$\sqrt{\pi/2} \cdot \left(\alpha \cdot \sqrt{2/\pi} \cdot +0 \right)$$

$$= \alpha = \langle \vec{q}, \vec{k} \rangle.$$

Variance analysis is also straightforward. Recall that:

$$\mathsf{X}_j = \sqrt{\pi/2}\, \langle \boldsymbol{\pi}_j, \vec{q} \rangle\, \mathsf{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

Squaring removes the sign, so

$$\mathbb{E}[\mathsf{X}_j^2] = \frac{\pi}{2} \cdot \mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{q} \rangle^2] = \frac{\pi}{2} \cdot \|\vec{q}\|_2^2.$$

QJL Expectation Analysis

$$\mathbb{E}[\sqrt{\pi/2} \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)]$$

$$\sqrt{\pi/2} \cdot \left(\alpha \cdot \sqrt{2/\pi} \cdot +0 \right)$$

$$= \alpha = \langle \vec{q}, \vec{k} \rangle.$$

Variance analysis is also straightforward. Recall that:

$$\mathsf{X}_j = \sqrt{\pi/2} \, \langle \boldsymbol{\pi}_j, \vec{q} \rangle \, \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

Squaring removes the sign, so

$$\mathbb{E}[\mathsf{X}_j^2] = \frac{\pi}{2} \cdot \mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{q} \rangle^2] = \frac{\pi}{2} \cdot \|\vec{q}\|_2^2.$$

$$\text{Var}[\mathsf{X}_j] = E[\mathsf{X}_j^2] - E[\mathsf{X}_j]^2 = \frac{\pi}{2} \cdot \|\vec{q}\|_2^2 - \alpha^2 \leq \frac{\pi}{2} \cdot \|\vec{q}\|_2^2.$$

QJL Expectation Analysis

$$\mathbb{E}[\sqrt{\pi/2} \cdot \langle \boldsymbol{\pi}_j, \vec{q} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)] = \mathbb{E}[\alpha \cdot |\langle \boldsymbol{\pi}_j, \vec{k} \rangle| + \langle \boldsymbol{\pi}_j, \vec{e} \rangle \cdot \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle)]$$

$$\sqrt{\pi/2} \cdot \left(\alpha \cdot \sqrt{2/\pi} \cdot +0 \right)$$

$$= \alpha = \langle \vec{q}, \vec{k} \rangle.$$

Variance analysis is also straightforward. Recall that:

$$\mathsf{X}_j = \sqrt{\pi/2}\, \langle \boldsymbol{\pi}_j, \vec{q} \rangle \, \text{sign}(\langle \boldsymbol{\pi}_j, \vec{k} \rangle).$$

Squaring removes the sign, so

$$\mathbb{E}[\mathsf{X}_j^2] = \frac{\pi}{2} \cdot \mathbb{E}[\langle \boldsymbol{\pi}_j, \vec{q} \rangle^2] = \frac{\pi}{2} \cdot \|\vec{q}\|_2^2.$$

$$\text{Var}[\mathsf{X}_j] = E[\mathsf{X}_j^2] - E[\mathsf{X}_j]^2 = \frac{\pi}{2} \cdot \|\vec{q}\|_2^2 - \alpha^2 \leq \frac{\pi}{2} \cdot \|\vec{q}\|_2^2.$$

The X_j are independent, so their average (i.e., the QJL estimator) has variance:

Data Dependent Dimensionality Reduction

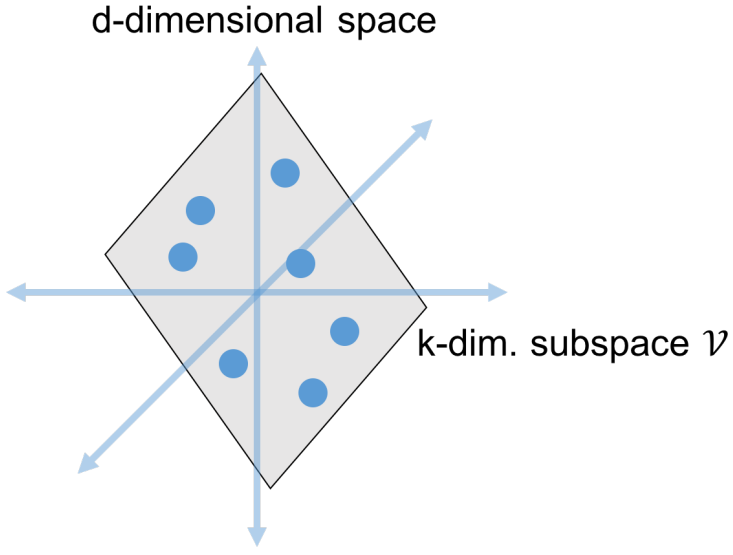
Data Dependent Dimensionality Reduction

- The JL Lemma is very general – it applies to any input data set.
- The compression Π is linear and data-oblivious. This is very useful for computational efficiency, and leads to applications in streaming and distributed settings.
- However, often we can exploit properties about the underlying input data to get more accurate compressions than what JL gives.
- Leads to a variety of data dependent dimensionality reduction methods.



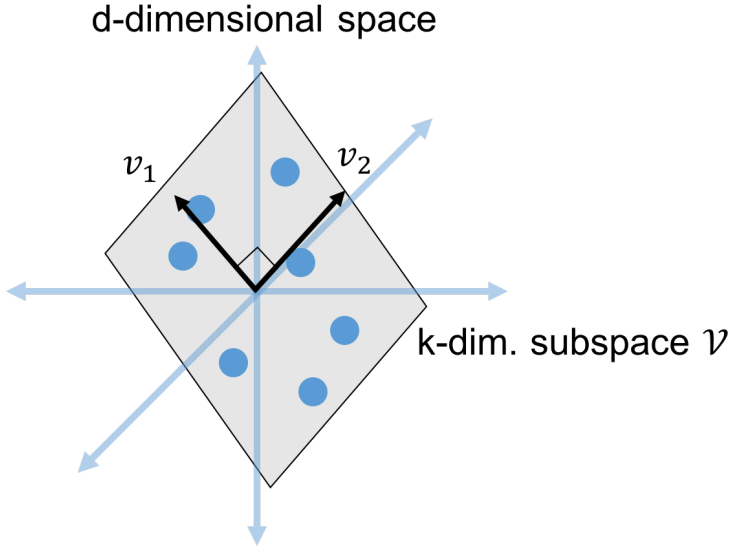
Embedding with Assumptions

Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Embedding with Assumptions

Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.

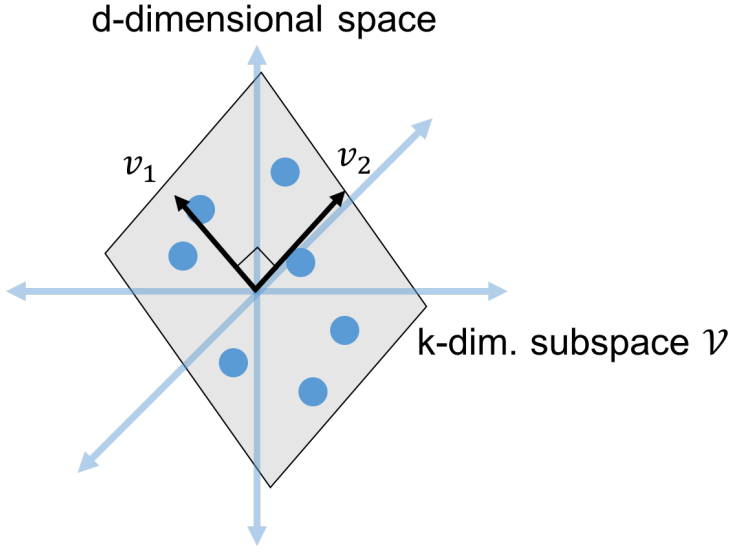


Claim: Let $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns. For all $\vec{x}_i, \vec{x}_j$:

$$\|\mathbf{V}^T\vec{x}_i - \mathbf{V}^T\vec{x}_j\|_2 = \|\vec{x}_i - \vec{x}_j\|_2.$$

Embedding with Assumptions

Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie in any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Claim: Let $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns. For all $\vec{x}_i, \vec{x}_j$:

$$\|\mathbf{V}^T\vec{x}_i - \mathbf{V}^T\vec{x}_j\|_2 = \|\vec{x}_i - \vec{x}_j\|_2.$$

- $\mathbf{V}^T \in \mathbb{R}^{k \times d}$ is a linear embedding of $\vec{x}_1, \ldots, \vec{x}_n$ into k dimensions with no distortion.

Very Simple Example

Another Example

Dot Product Transformation

Claim: Let $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns. For all $\vec{x}_i, \vec{x}_j \in \mathcal{V}$:

$$\|\mathbf{V}^T \vec{x}_i - \mathbf{V}^T \vec{x}_j\|_2 = \|\vec{x}_i - \vec{x}_j\|_2.$$

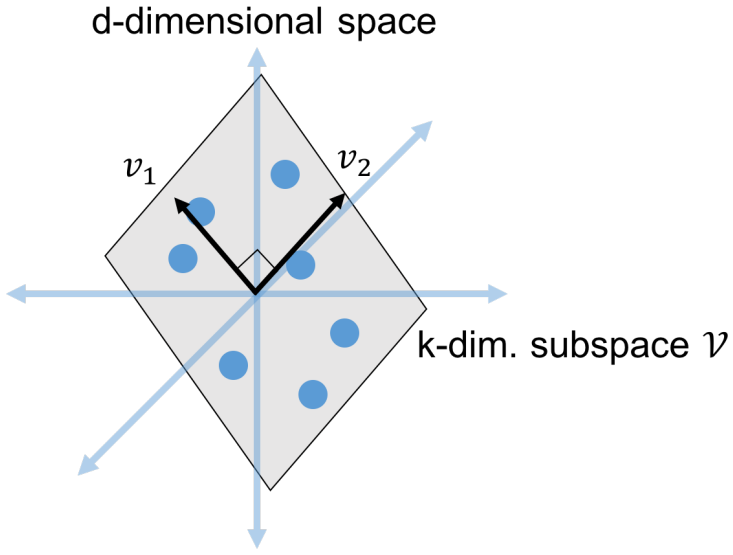
Dot Product Transformation

Claim: Let $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns. For all $\vec{x}_i, \vec{x}_j \in \mathcal{V}$:

$$\|\mathbf{V}^T \vec{x}_i - \mathbf{V}^T \vec{x}_j\|_2 = \|\vec{x}_i - \vec{x}_j\|_2.$$

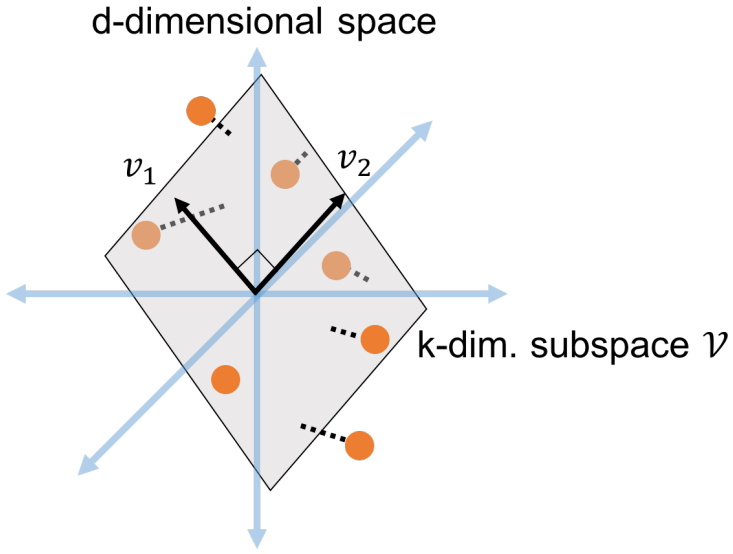
Embedding with Assumptions

Main Focus of Upcoming Classes: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



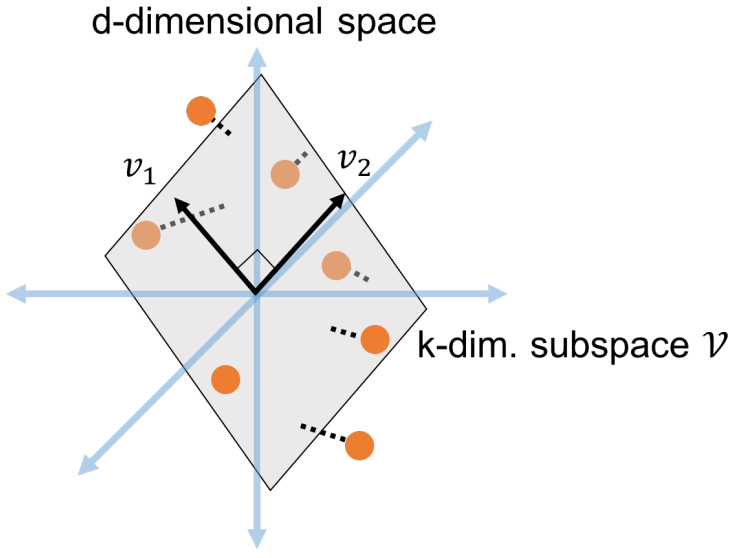
Embedding with Assumptions

Main Focus of Upcoming Classes: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Embedding with Assumptions

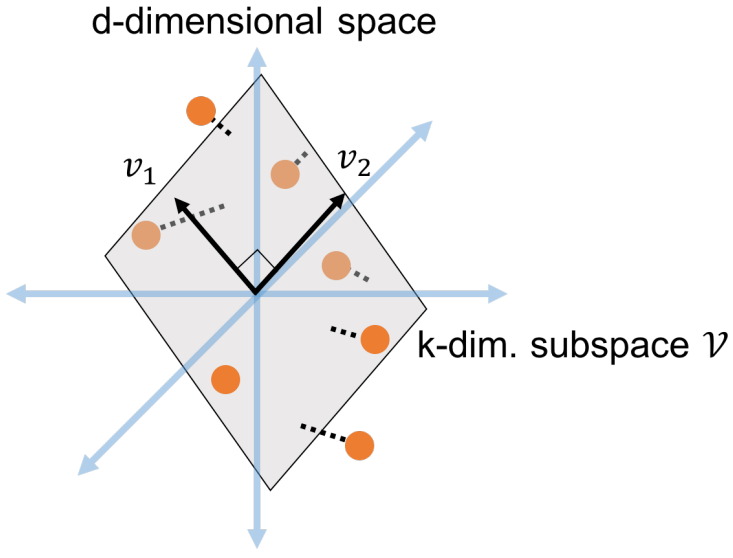
Main Focus of Upcoming Classes: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Letting $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns, $\mathbf{V}^T \vec{x}_i \in \mathbb{R}^k$ is still a good embedding for $x_i \in \mathbb{R}^d$.

Embedding with Assumptions

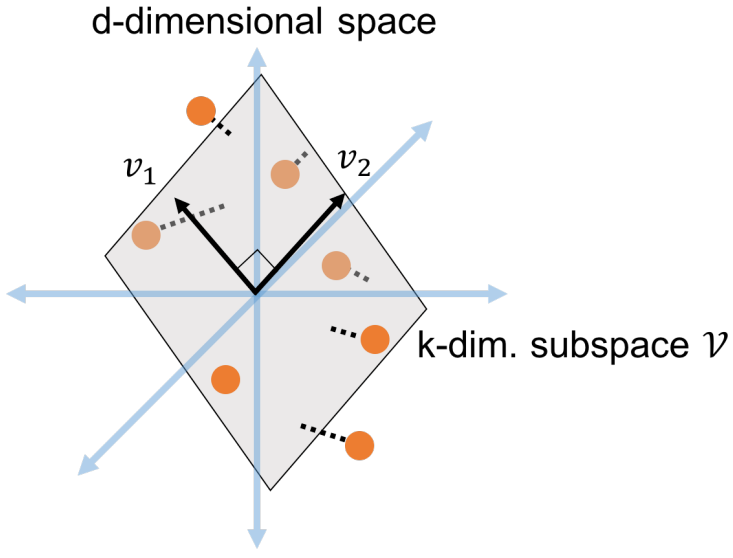
Main Focus of Upcoming Classes: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Letting $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns, $\mathbf{V}^T \vec{x}_i \in \mathbb{R}^k$ is still a good embedding for $x_i \in \mathbb{R}^d$. The key idea behind low-rank approximation and principal component analysis (PCA).

Embedding with Assumptions

Main Focus of Upcoming Classes: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$.



Letting $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns, $\mathbf{V}^T \vec{x}_i \in \mathbb{R}^k$ is still a good embedding for $x_i \in \mathbb{R}^d$. The key idea behind low-rank approximation and principal component analysis (PCA).

- How do we find $\mathcal{V}$ and $\mathbf{V}$?
- How good is the embedding?

Low-Rank Factorization

Claim: $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

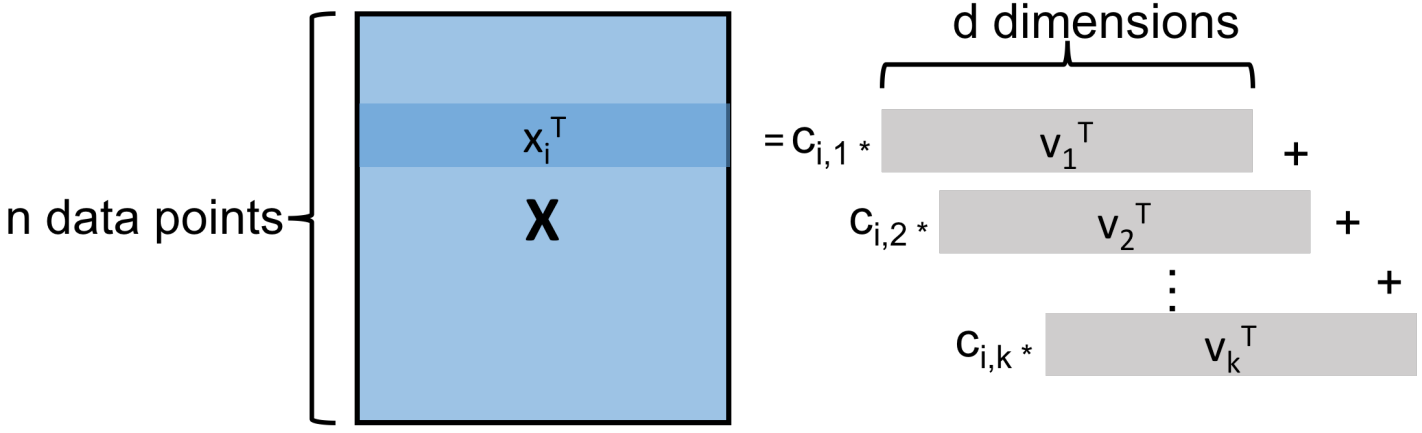
$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

Claim: $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

- Letting $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$, can write $\vec{x}_i$ as:

$$\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + c_{i,2} \cdot \vec{v}_2 + \ldots + c_{i,k} \cdot \vec{v}_k.$$



$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

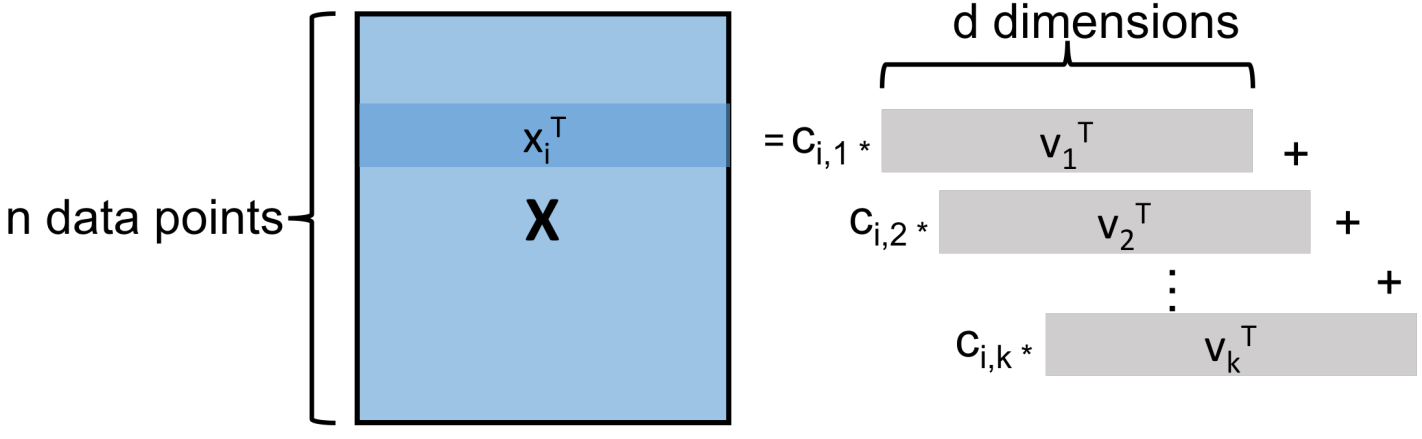
Low-Rank Factorization

Claim: $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

- Letting $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$, can write $\vec{x}_i$ as:

$$\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + c_{i,2} \cdot \vec{v}_2 + \ldots + c_{i,k} \cdot \vec{v}_k.$$

- So $\vec{v}_1, \ldots, \vec{v}_k$ span the rows of $\mathbf{X}$ and thus $\text{rank}(\mathbf{X}) \leq k$.



$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

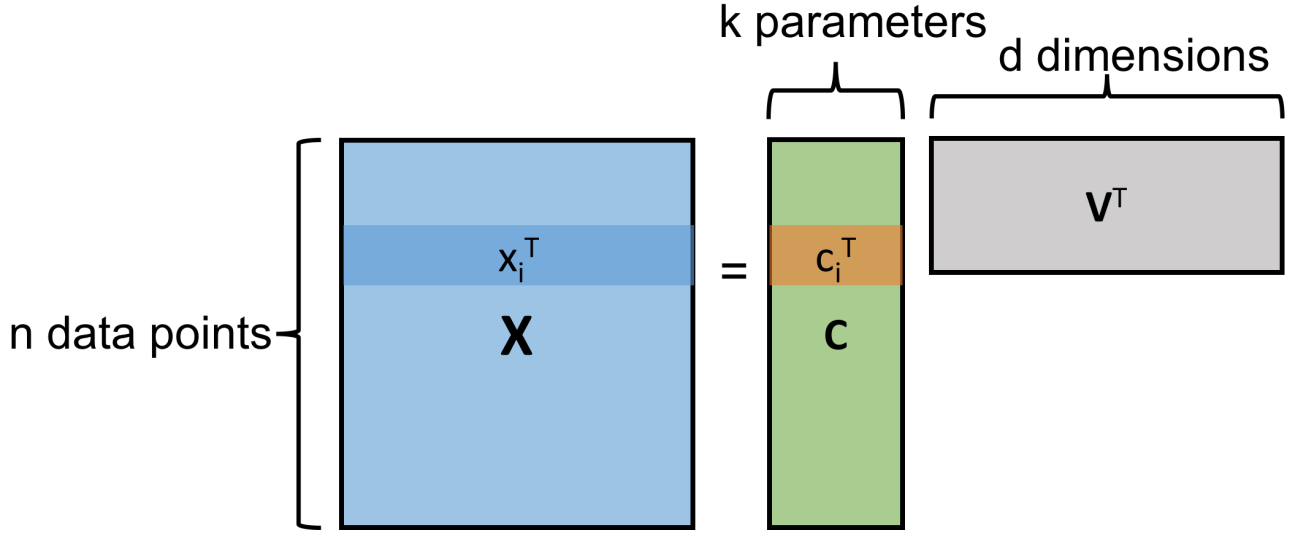
Claim: $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

- Every data point $\vec{x}_i$ (row of $\mathbf{X}$) can be written as $\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + \ldots + c_{i,k} \cdot \vec{v}_k$.

$\vec{x}_1, \ldots, \vec{x}_n$: data points (in $\mathbb{R}^d$), $\mathcal{V}$: k-dimensional subspace of $\mathbb{R}^d$, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Claim: $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

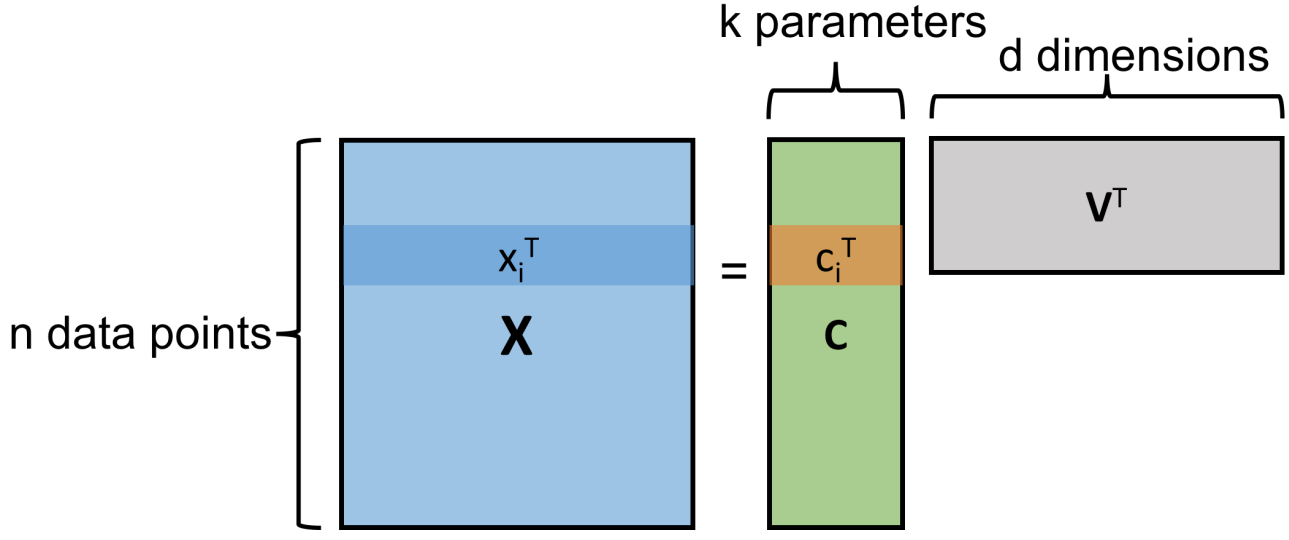
- Every data point $\vec{x}_i$ (row of $\mathbf{X}$) can be written as $\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + \ldots + c_{i,k} \cdot \vec{v}_k$.



$\vec{x}_1, \ldots, \vec{x}_n$: data points (in $\mathbb{R}^d$), $\mathcal{V}$: k-dimensional subspace of $\mathbb{R}^d$, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Claim: $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

- Every data point $\vec{x}_i$ (row of $\mathbf{X}$) can be written as $\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + \ldots + c_{i,k} \cdot \vec{v}_k$.

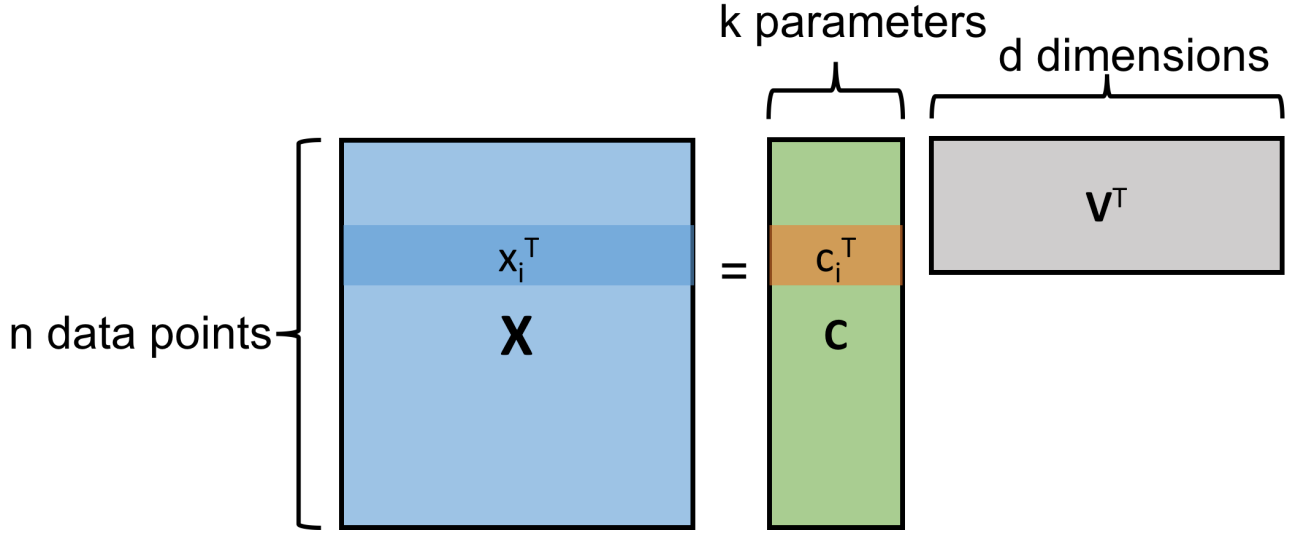


- $\mathbf{X}$ can be represented by $(n + d) \cdot k$ parameters vs. $n \cdot d$.

$\vec{x}_1, \ldots, \vec{x}_n$: data points (in $\mathbb{R}^d$), $\mathcal{V}$: k-dimensional subspace of $\mathbb{R}^d$, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Claim: $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ lie in a k-dimensional subspace $\mathcal{V} \Leftrightarrow$ the data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has rank $\leq k$.

- Every data point $\vec{x}_i$ (row of $\mathbf{X}$) can be written as $\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + \ldots + c_{i,k} \cdot \vec{v}_k$.

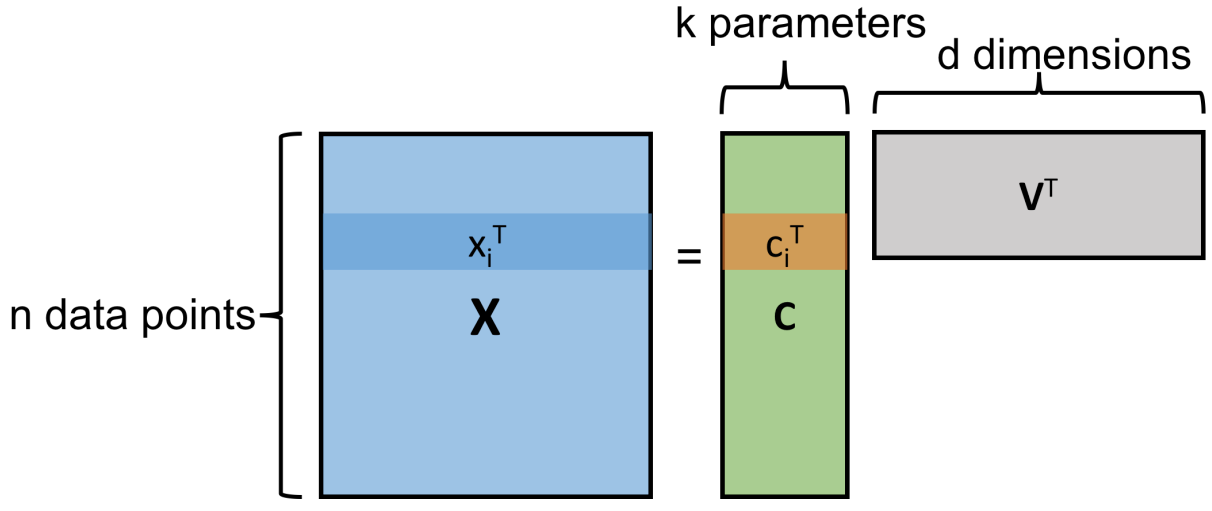


- $\mathbf{X}$ can be represented by $(n + d) \cdot k$ parameters vs. $n \cdot d$.
- The rows of $\mathbf{X}$ are spanned by k vectors: the columns of $\mathbf{V} \implies$ the columns of $\mathbf{X}$ are spanned by k vectors: the columns of $\mathbf{C}$.

$\vec{x}_1, \ldots, \vec{x}_n$: data points (in $\mathbb{R}^d$), $\mathcal{V}$: k-dimensional subspace of $\mathbb{R}^d$, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

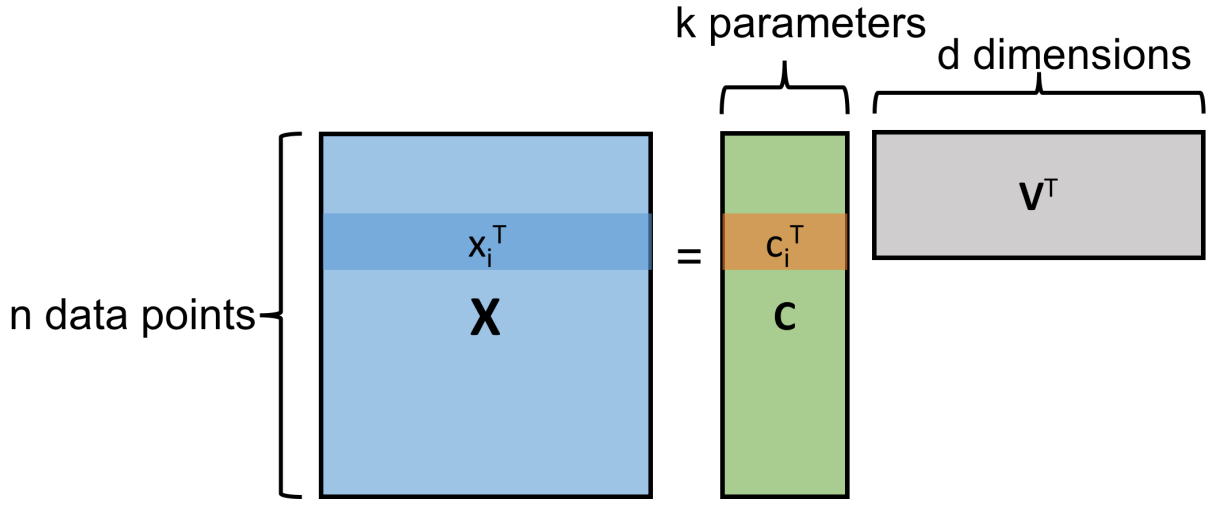
Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.



$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.

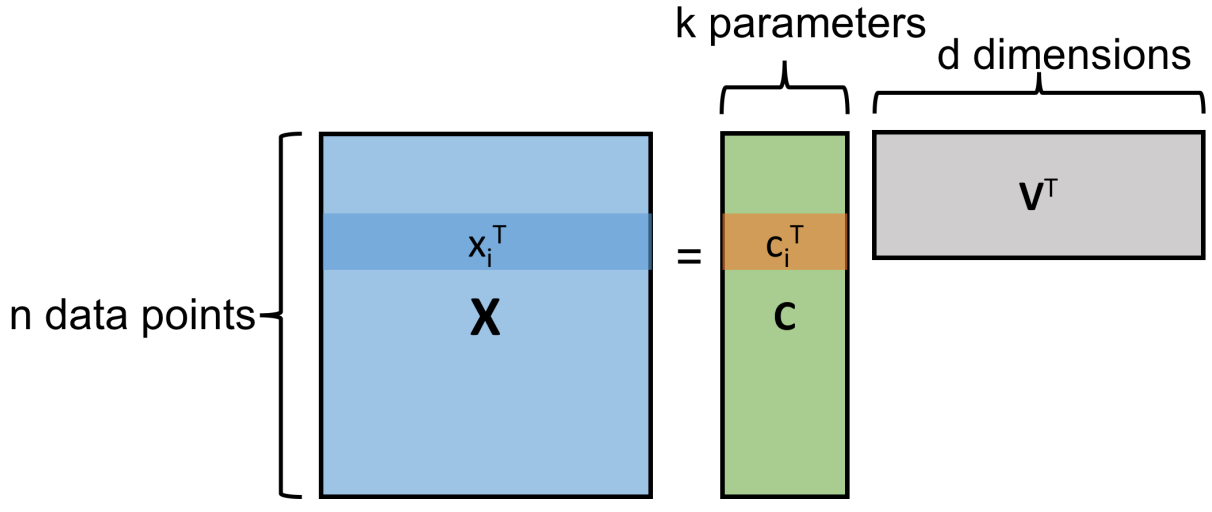


Exercise: What is this coefficient matrix $\mathbf{C}$? **Hint:** Use that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.



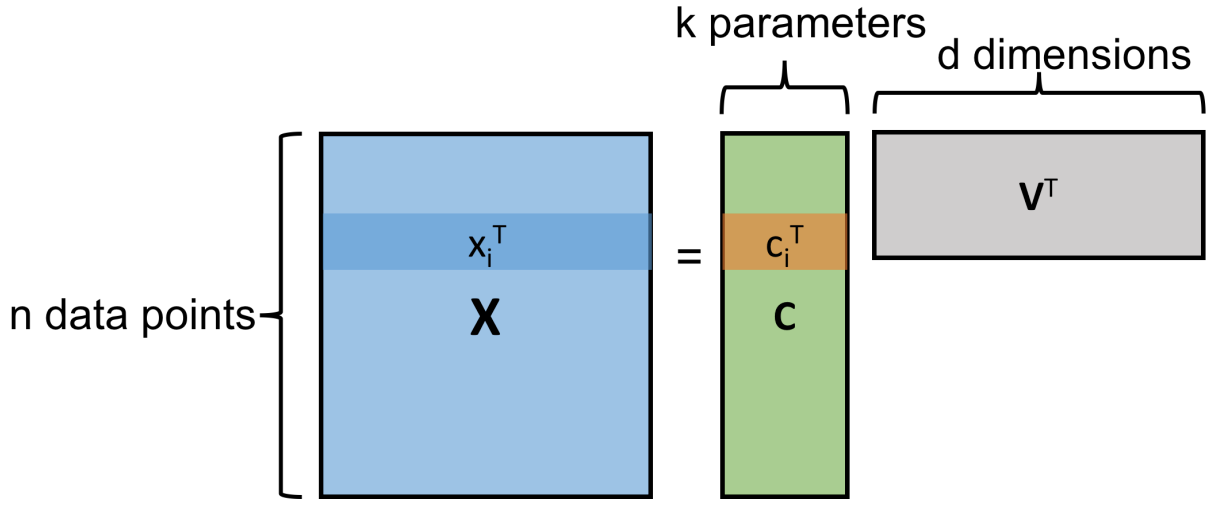
Exercise: What is this coefficient matrix $\mathbf{C}$? **Hint:** Use that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

- $\mathbf{X} = \mathbf{C}\mathbf{V}^T \implies \mathbf{X}\mathbf{V} = \mathbf{C}\mathbf{V}^T\mathbf{V}$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.



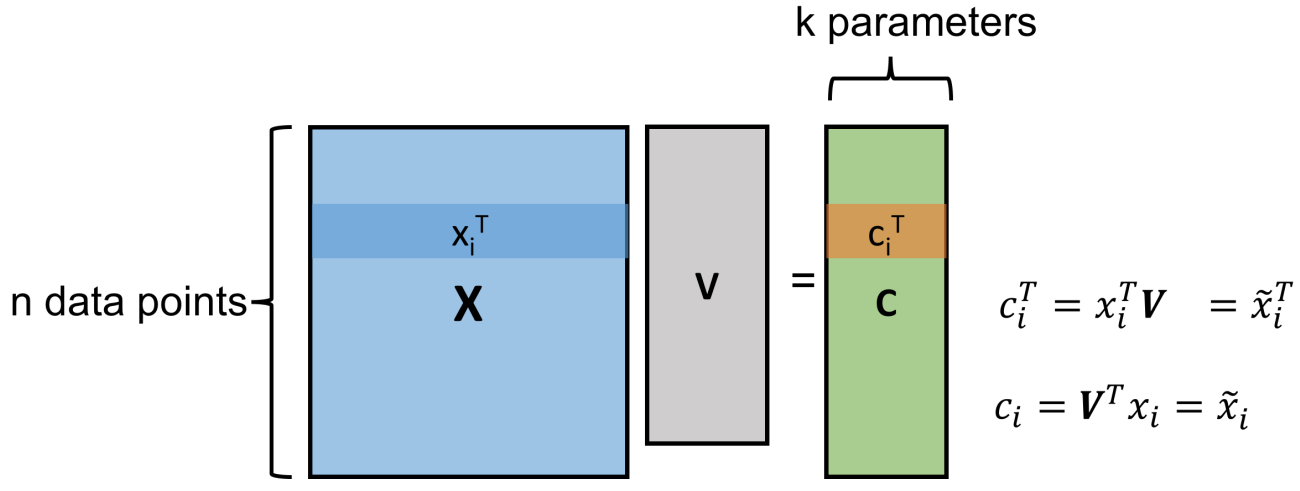
Exercise: What is this coefficient matrix $\mathbf{C}$? **Hint:** Use that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

- $\mathbf{X} = \mathbf{C}\mathbf{V}^T \implies \mathbf{X}\mathbf{V} = \mathbf{C}\mathbf{V}^T\mathbf{V} \implies \mathbf{X}\mathbf{V} = \mathbf{C}$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Factorization

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.



Exercise: What is this coefficient matrix $\mathbf{C}$? **Hint:** Use that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

- $\mathbf{X} = \mathbf{C}\mathbf{V}^T \implies \mathbf{X}\mathbf{V} = \mathbf{C}\mathbf{V}^T\mathbf{V} \implies \mathbf{X}\mathbf{V} = \mathbf{C}$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{C}\mathbf{V}^T.$$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

- $\mathbf{V}\mathbf{V}^T$ is a projection matrix, which projects vectors onto the subspace $\mathcal{V}$.

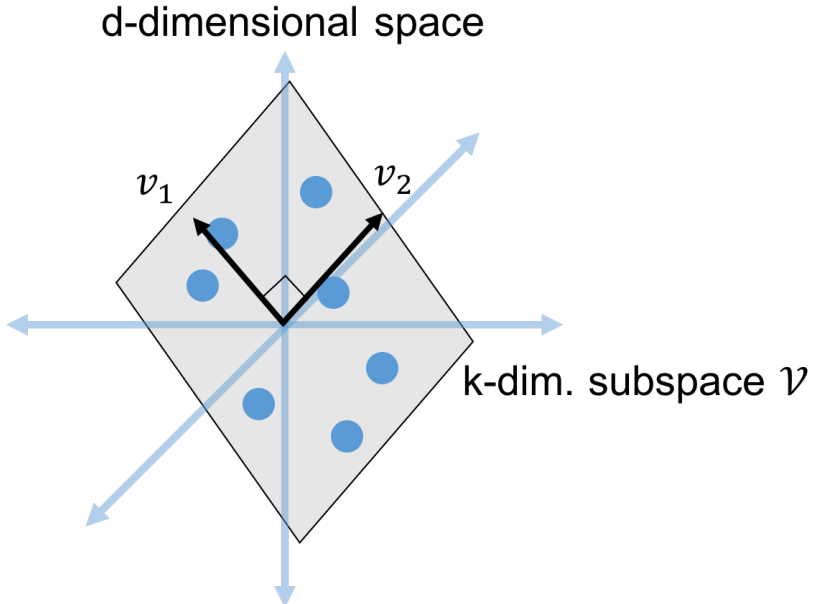
$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

- $\mathbf{V}\mathbf{V}^T$ is a projection matrix, which projects vectors onto the subspace $\mathcal{V}$.



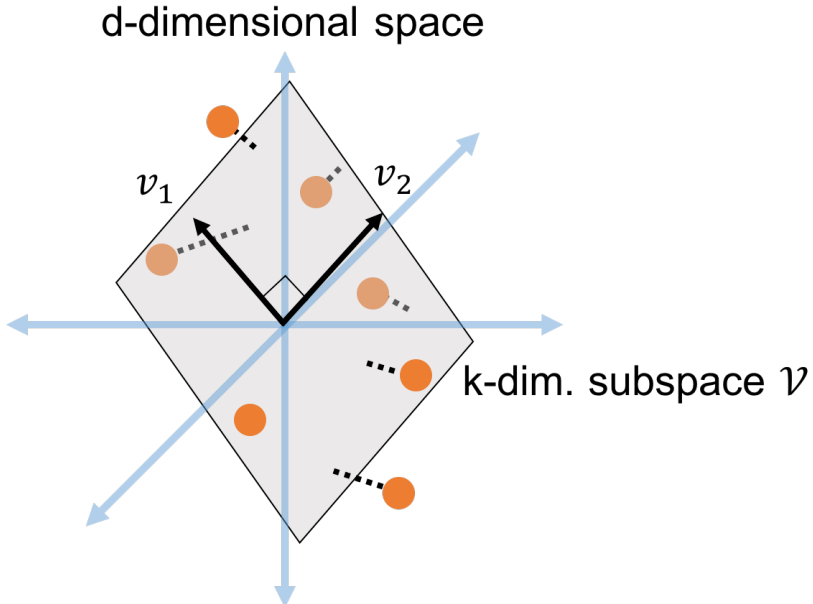
$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

- $\mathbf{V}\mathbf{V}^T$ is a projection matrix, which projects vectors onto the subspace $\mathcal{V}$.



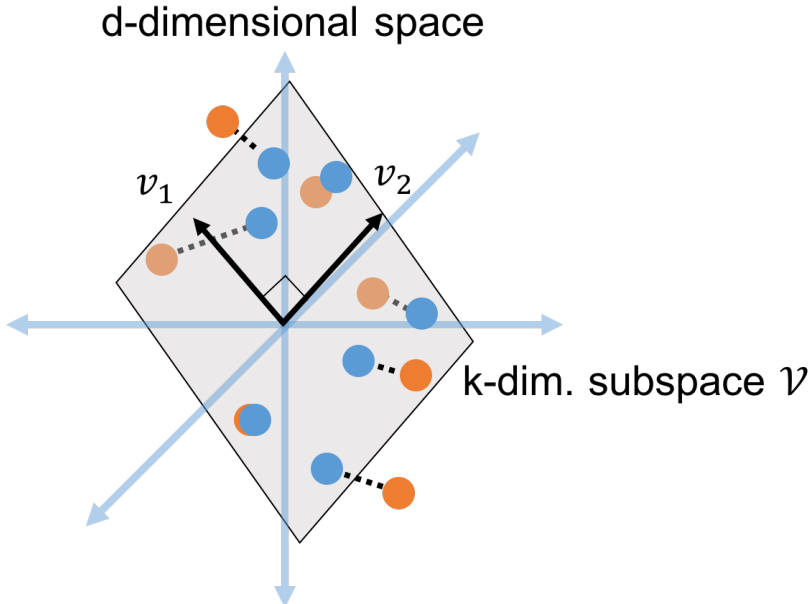
$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie in a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be written as

$$\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

- $\mathbf{V}\mathbf{V}^T$ is a projection matrix, which projects vectors onto the subspace $\mathcal{V}$.

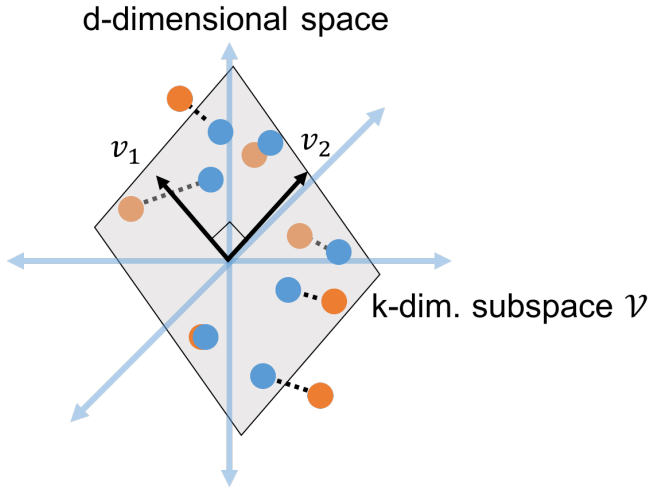


$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T$$

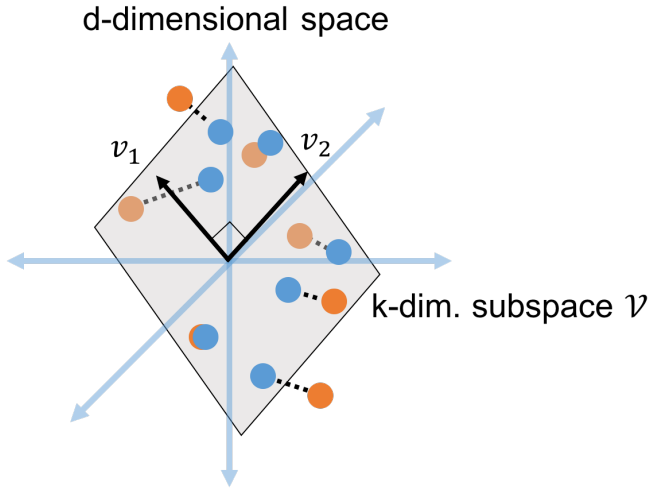


$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T$$



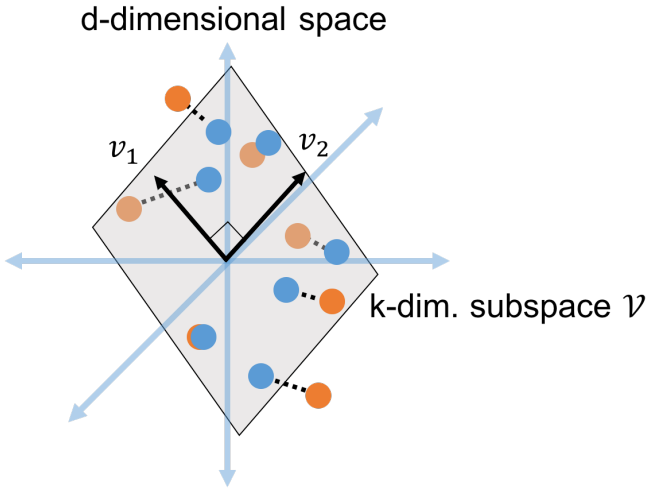
Note: $\mathbf{X}\mathbf{V}\mathbf{V}^T$ has rank k. It is a low-rank approximation of $\mathbf{X}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

Claim: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T$$



Note: $\mathbf{X}\mathbf{V}\mathbf{V}^T$ has rank k. It is a low-rank approximation of $\mathbf{X}$.

$$\mathbf{X}\mathbf{V}\mathbf{V}^\mathsf{T} = \underset{\mathbf{B} \text{ with rows in } \mathcal{V}}{\arg\min} \|\mathbf{X} - \mathbf{B}\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - \mathbf{B}_{i,j})^2.$$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

So Far: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

This is the closest approximation to $\mathbf{X}$ with rows in $\mathcal{V}$ (i.e., in the column span of $\mathbf{V}$).

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

So Far: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

This is the closest approximation to $\mathbf{X}$ with rows in $\mathcal{V}$ (i.e., in the column span of $\mathbf{V}$).

- Letting $(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i, (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j$ be the i^{th} and j^{th} projected data points,

$$\|(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i - (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\mathbf{V}^T\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\|_2.$$

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

So Far: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

This is the closest approximation to $\mathbf{X}$ with rows in $\mathcal{V}$ (i.e., in the column span of $\mathbf{V}$).

- Letting $(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i, (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j$ be the i^{th} and j^{th} projected data points,

$$\|(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i - (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\mathbf{V}^T\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\|_2.$$

- Can use $\mathbf{X}\mathbf{V} \in \mathbb{R}^{n \times k}$ as a compressed approximate data set.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation

So Far: If $\vec{x}_1, \ldots, \vec{x}_n$ lie close to a k-dimensional subspace $\mathcal{V}$ with orthonormal basis $\mathbf{V} \in \mathbb{R}^{d \times k}$, the data matrix can be approximated as:

$$\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T.$$

This is the closest approximation to $\mathbf{X}$ with rows in $\mathcal{V}$ (i.e., in the column span of $\mathbf{V}$).

- Letting $(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i, (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j$ be the i^{th} and j^{th} projected data points,

$$\|(\mathbf{X}\mathbf{V}\mathbf{V}^T)_i - (\mathbf{X}\mathbf{V}\mathbf{V}^T)_j\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\mathbf{V}^T\|_2 = \|[(\mathbf{X}\mathbf{V})_i - (\mathbf{X}\mathbf{V})_j]\|_2.$$

- Can use $\mathbf{X}\mathbf{V} \in \mathbb{R}^{n \times k}$ as a compressed approximate data set.

Key question is how to find the subspace $\mathcal{V}$ and correspondingly $\mathbf{V}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Properties of Projection Matrices

Quick Exercise: Show that $\mathbf{V}\mathbf{V}^T$ is idempotent. I.e., $(\mathbf{V}\mathbf{V}^T)(\mathbf{V}\mathbf{V}^T)\vec{y} = (\mathbf{V}\mathbf{V}^T)\vec{y}$ for any $\vec{y} \in \mathbb{R}^d$.

Why does this make sense intuitively?

Less Quick Exercise: (Pythagorean Theorem) Show that:

$$\|\vec{y}\|_2^2 = \|(\mathbf{V}\mathbf{V}^T)\vec{y}\|_2^2 + \|\vec{y} - (\mathbf{V}\mathbf{V}^T)\vec{y}\|_2^2.$$