# COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2023.

Lecture 15

# Logistics

- Midterm grades and solutions are posted on Moodle.
- We'll hand out the midterms at the end of class.
- The class average was $\approx 30/39 = 77\%$.
- See Piazza post for more details. If you aren't happy with your grade, I'm happy to chat about strategies moving forward.

# Quiz Question

**Question 5**

Not complete

Points out of 1.00

Flag question

Edit question

Suppose x=(1,2,3,4) and let y=($y_1$,$y_2$,$y_3$,$y_4$) be a random vector where each $y_i$ is independent and is distributed according to a Normal distribution with mean 0 and variance 1. What is the expected value of $\langle x, y \rangle^2$?

Answer:

Check

# Summary

**Last Few Classes:** The Johnson-Lindenstrauss Lemma

- Reduce $n$ data points in any dimension $d$ to $O\left(\frac{\log n/\delta}{\epsilon^2}\right)$ dimensions and preserve (with probability $\geq 1-\delta$) all pairwise distances up to $1 \pm \epsilon$.
- Compression is linear via multiplication with a random, data oblivious, matrix (linear compression)
- Proved via the distributional JL-Lemma which shows that if $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ is a random matrix, $\mathbf{\Pi}\vec{y}_2 \approx \|\vec{y}\|$ for any $y$ with high probability.
- Proof of distributional JL via linearity of expectation, linearity of variance, stability of the Gaussian distribution, and an exponential concentration bound for Chi-Squared random variables.

# Summary

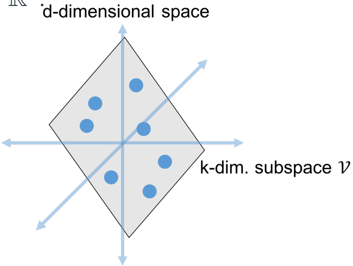
Next Few Classes: Low-rank approximation, the SVD, and principal component analysis (PCA).

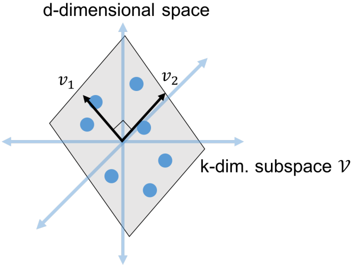
- Reduce $d$-dimesional data points to a smaller dimension $m$.
- Like JL, compression is linear – by applying a matrix.
- Chose this matrix carefully, taking into account structure of the dataset.
- Can give better compression than random projection (although not directly comparable).

Will be using a fair amount of linear algebra: orthogonal basis, column/row span, eigenvectors, etc.

# 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.

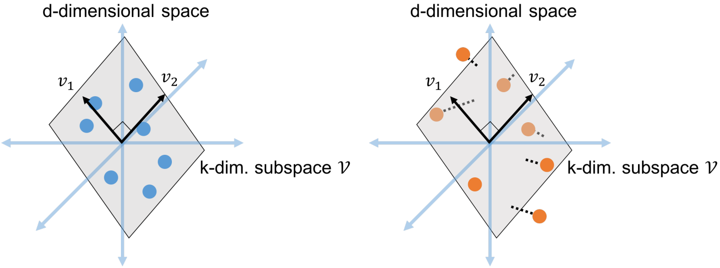
# Dot Product Transformation

**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 $\mathbb{R}^d$.




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?

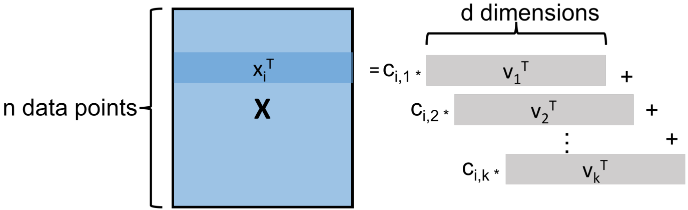
# Low-Rank Factorization

**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 $\leq k$.

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

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

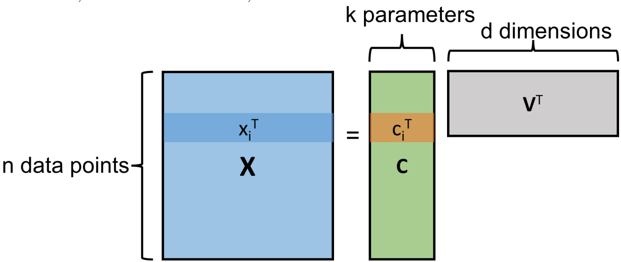
- So $\vec{v}_1, \ldots, \vec{v}_k$ span the rows of $\mathbf{X}$ and thus $\mathbf{rank}(\mathbf{X}) \leq k$.




$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: 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 $\vec{x}_i = \mathbf{V}\vec{c}_i = c_{i,1} \cdot \vec{v}_1 + \ldots + c_{i,k} \cdot \vec{v}_k$.

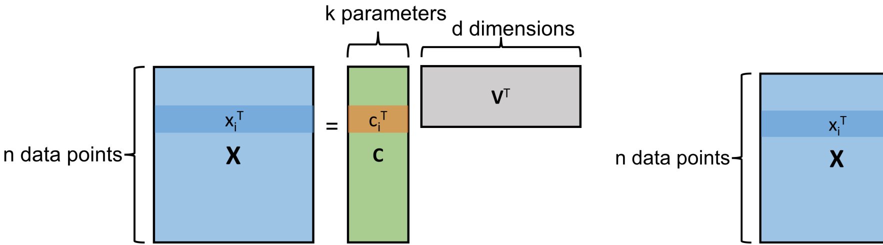



- $\mathbf{X}$ can be represented by $(n + d) \cdot k$ parameters vs. $n \cdot d$.
- The rows of $\mathbf{X}$ are spanned by $k$ vectors: the columns of $\mathbf{V}$ $\implies$ the columns of $\mathbf{X}$ are spanned by $k$ vectors: the columns of $\mathbf{C}$.

$\vec{x}_1, \ldots, \vec{x}_n$: 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 $\mathbf{X} = \mathbf{C}\mathbf{V}^T$.




**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 $\vec{v}_1, \ldots, \vec{v}_k$.

**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

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

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