

COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Spring 2026.

Lecture 12

- Quiz this week due Monday at 8pm.
- Several survey style questions. Due to Canvas limitations, some options may be marked incorrect but still will receive full credit.
- So just ignore the grading on the survey style questions.

Summary

Last Class:

- Intro to dimensionality reduction.
- Low-distortion embeddings and the Johnson-Lindenstrauss (JL) Lemma.
- Most of the proof of the JL lemma.

This Class:

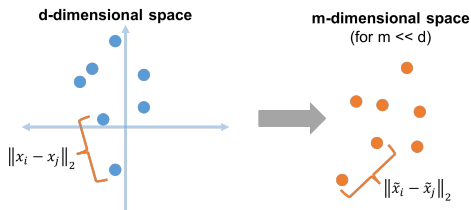
- Finish JL Lemma proof.
- Example application to clustering.
- Start on data dependent dimensionality reduction.

The Johnson-Lindenstrauss Lemma

Johnson-Lindenstrauss Lemma: For any set of points $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$ and $\epsilon > 0$ there exists a linear map $\mathbf{\Pi} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ such that $m = O\left(\frac{\log n}{\epsilon^2}\right)$ and letting $\tilde{x}_i = \mathbf{\Pi}\vec{x}_i$:

For all i, j : $(1 - \epsilon)\|\vec{x}_i - \vec{x}_j\|_2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2 \leq (1 + \epsilon)\|\vec{x}_i - \vec{x}_j\|_2$.

Further, if $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ has each entry chosen i.i.d. from $\mathcal{N}(0, 1/m)$, it satisfies the guarantee with high probability.



Distributional JL

The Johnson-Lindenstrauss Lemma is a direct consequence of a closely related lemma:

Distributional JL Lemma: Let $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$. If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, then for any $\vec{y} \in \mathbb{R}^d$, with probability $\geq 1 - \delta$

$$(1 - \epsilon)\|\vec{y}\|_2 \leq \|\mathbf{\Pi}\vec{y}\|_2 \leq (1 + \epsilon)\|\vec{y}\|_2$$

Applying a random matrix $\mathbf{\Pi}$ to any vector \vec{y} preserves \vec{y} 's norm with high probability.

Distributional JL Lemma \implies JL Lemma: Apply Distributional-JL to $y_{ij} = x_i - x_j$ for all $\binom{n}{2}$ pairs x_i, x_j . Then take a union bound to say distance is preserved between all pairs with high probability.

Distributional JL Proof

Letting $\mathbf{\Pi} \in \mathbb{R}^{d \times m}$ have each entry chosen i.i.d. as $\mathcal{N}(0, 1/m)$, for any $\vec{y} \in \mathbb{R}^d$, letting $\tilde{\mathbf{y}} = \mathbf{\Pi}\vec{y}$:

$$\tilde{y}(j) \sim \mathcal{N}(0, \|\vec{y}\|_2^2/m) \text{ and } \mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \|\vec{y}\|_2^2$$

$\|\tilde{\mathbf{y}}\|_2^2 = \sum_{j=1}^m \tilde{y}(j)^2$ a **Chi-Squared random variable with m degrees of freedom** (a sum of m squared independent Gaussians).

Lemma: (Chi-Squared Concentration) Letting Z be a Chi-Squared random variable with m degrees of freedom,

$$\Pr[|Z - \mathbb{E}Z| \geq \epsilon \mathbb{E}Z] \leq 2e^{-m\epsilon^2/8}.$$

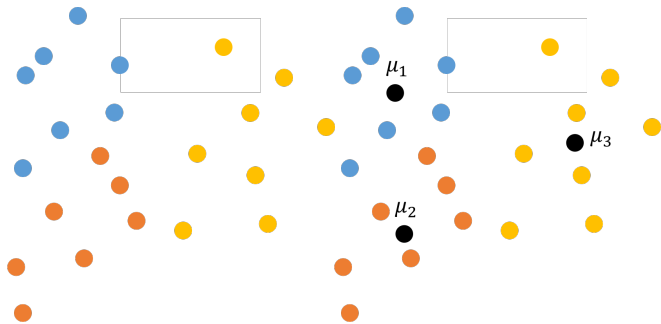
If we set $m = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, with probability $1 - O(e^{-\log(1/\delta)}) \geq 1 - \delta$:

$$(1 - \epsilon)\|\vec{y}\|_2^2 \leq \|\tilde{\mathbf{y}}\|_2^2 \leq (1 + \epsilon)\|\vec{y}\|_2^2.$$

Gives the distributional JL Lemma and thus the classic JL Lemma.

Example Application: k -means clustering

Goal: Separate n points in d dimensional space into k groups.



k-means Objective: $Cost(\mathcal{C}_1, \dots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \dots, \mathcal{C}_k} \sum_{j=1}^k \sum_{\vec{x} \in \mathcal{C}_k} \|\vec{x} - \mu_j\|_2^2$.

Write in terms of distances:

$$Cost(\mathcal{C}_1, \dots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \dots, \mathcal{C}_k} \sum_{j=1}^k \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$$

Example Application: k -means clustering

k-means Objective: $Cost(\mathcal{C}_1, \dots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \dots, \mathcal{C}_k} \sum_{j=1}^k \sum_{\vec{x}_1, \vec{x}_2 \in \mathcal{C}_k} \|\vec{x}_1 - \vec{x}_2\|_2^2$

If we randomly project to $m = O\left(\frac{\log n}{\epsilon^2}\right)$ dimensions, for all pairs \vec{x}_1, \vec{x}_2 ,

$$(1 - \epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \leq \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2 \leq (1 + \epsilon)\|\vec{x}_1 - \vec{x}_2\|_2^2 \implies$$

Letting $\overline{Cost}(\mathcal{C}_1, \dots, \mathcal{C}_k) = \min_{\mathcal{C}_1, \dots, \mathcal{C}_k} \sum_{j=1}^k \sum_{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{C}_k} \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2\|_2^2$

$$(1 - \epsilon)Cost(\mathcal{C}_1, \dots, \mathcal{C}_k) \leq \overline{Cost}(\mathcal{C}_1, \dots, \mathcal{C}_k) \leq (1 + \epsilon)Cost(\mathcal{C}_1, \dots, \mathcal{C}_k).$$

Upshot: Can cluster in m dimensional space (much more efficiently) and minimize $\overline{Cost}(\mathcal{C}_1, \dots, \mathcal{C}_k)$. The optimal set of clusters will have true cost within $\frac{1+\epsilon}{1-\epsilon} \approx 1 + 2\epsilon$ times the true optimal. **Good exercise to prove this.**

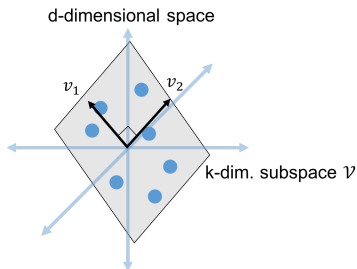
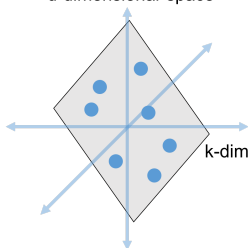
Data Dependent Dimensionality Reduction

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



Claim: Let $\vec{v}_1, \dots, \vec{v}_k$ 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 $\vec{x}_1, \dots, \vec{x}_n$ into k dimensions with **no distortion**.

Very Simple Example

Another Example

Dot Product Transformation

Claim: Let $\vec{v}_1, \dots, \vec{v}_k$ 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 $\vec{x}_i, \vec{x}_j \in \mathcal{V}$:

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

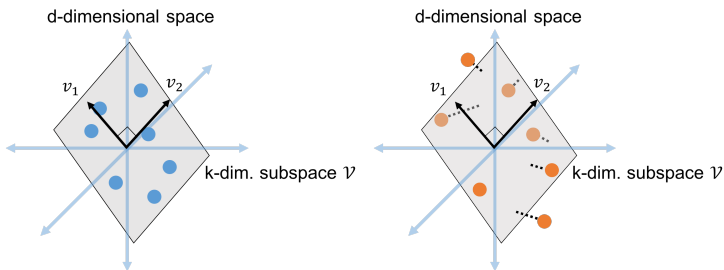
Dot Product Transformation

Claim: Let $\vec{v}_1, \dots, \vec{v}_k$ 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 $\vec{x}_1, \dots, \vec{x}_n$ lie **close to** any k -dimensional subspace \mathcal{V} of \mathbb{R}^d .



Letting $\vec{v}_1, \dots, \vec{v}_k$ 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?

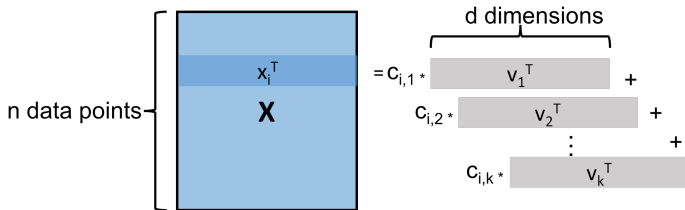
Low-Rank Factorization

Claim: $\vec{x}_1, \dots, \vec{x}_n$ 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 $\vec{v}_1, \dots, \vec{v}_k$ be an orthonormal basis for \mathcal{V} , can write \vec{x}_i as:

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

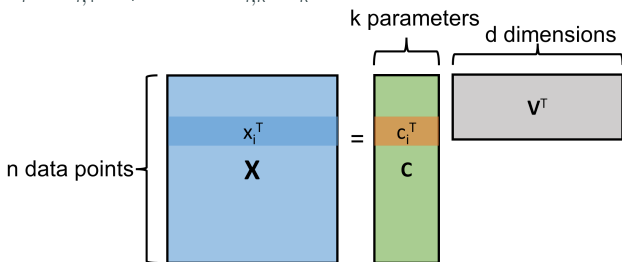
- So $\vec{v}_1, \dots, \vec{v}_k$ span the rows of \mathbf{X} and thus $\text{rank}(\mathbf{X}) \leq k$.



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

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

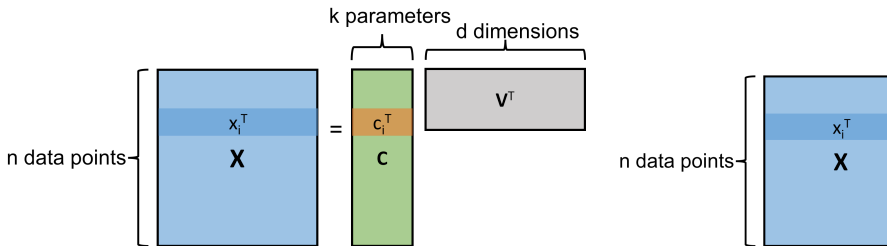


- \mathbf{X} can be represented by $(n + d) \cdot k$ parameters vs. $n \cdot d$.
- The rows of \mathbf{X} are spanned by k vectors: the columns of $\mathbf{V} \implies$ the columns of \mathbf{X} are spanned by k vectors: the columns of \mathbf{C} .

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

Low-Rank Factorization

Claim: If $\vec{x}_1, \dots, \vec{x}_n$ 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 $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

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

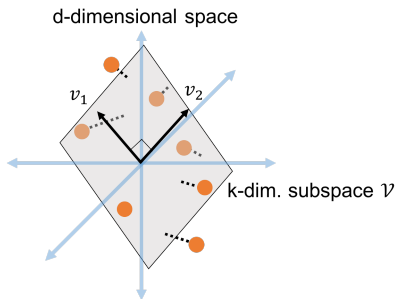
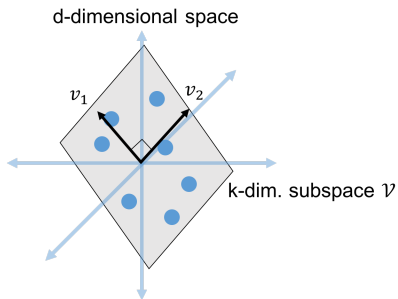
$\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \dots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace \mathcal{V} . $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \dots, \vec{v}_k$.

Projection View

Claim: If $\vec{x}_1, \dots, \vec{x}_n$ 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

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

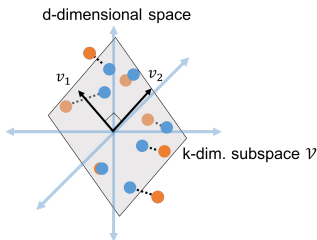
- \mathbf{V}^T is a **projection matrix**, which projects vectors onto the subspace \mathcal{V} .



Low-Rank Approximation

Claim: If $\vec{x}_1, \dots, \vec{x}_n$ 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:**

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



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

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

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

Low-Rank Approximation

So Far: If $\vec{x}_1, \dots, \vec{x}_n$ 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:

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

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

- Letting $(\mathbf{XV}^T)_i, (\mathbf{XV}^T)_j$ be the i^{th} and j^{th} projected data points,
$$\|(\mathbf{XV}^T)_i - (\mathbf{XV}^T)_j\|_2 = \|[(\mathbf{XV})_i - (\mathbf{XV})_j]\mathbf{V}^T\|_2 = \|[(\mathbf{XV})_i - (\mathbf{XV})_j]\|_2.$$
- Can use $\mathbf{XV} \in \mathbb{R}^{n \times k}$ as a compressed approximate data set.

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

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

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:

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