COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Spring 2020.

Lecture 15

Last Class: Low-Rank Approximation

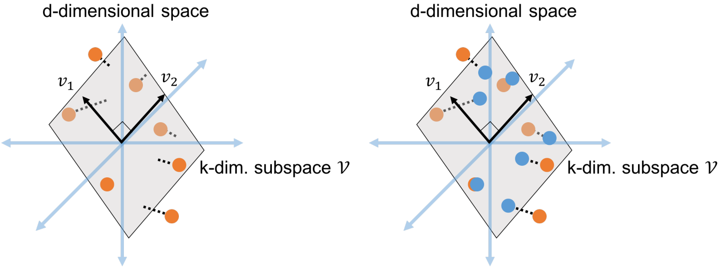
- When data lies in a k-dimensional subspace $\mathcal{V}$, we can perfectly embed into k dimensions using an orthonormal span $\mathbf{V} \in \mathbb{R}^{d \times k}$.
- When data lies close to $\mathcal{V}$, the optimal embedding in that space is given by projecting onto that space.

$$\mathbf{X}\mathbf{V}\mathbf{V}^T = \underset{\mathbf{B} \text{ with rows in } \mathcal{V}}{\arg\min} \|\mathbf{X} - \mathbf{B}\|_F^2.$$

This Class: Finding $\mathcal{V}$ via eigendecomposition.

- How do we find the best low-dimensional subspace to approximate $\mathbf{X}$?
- PCA and its connection to eigendecomposition.

Set Up: Assume that data points $\vec{x}_1, \ldots, \vec{x}_n$ lie close to any k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^d$. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the data matrix.

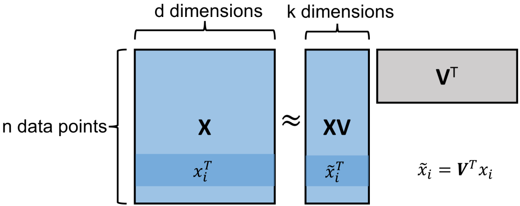


Let $\vec{v}_1, \ldots, \vec{v}_k$ be an orthonormal basis for $\mathcal{V}$ and $\mathbf{V} \in \mathbb{R}^{d \times k}$ be the matrix with these vectors as its columns.

- $\mathbf{V}\mathbf{V}^T \in \mathbb{R}^{d \times d}$ is the projection matrix onto $\mathcal{V}$.
- $\mathbf{X} \approx \mathbf{X}(\mathbf{V}\mathbf{V}^T)$. Gives the closest approximation to $\mathbf{X}$ with rows in $\mathcal{V}$.

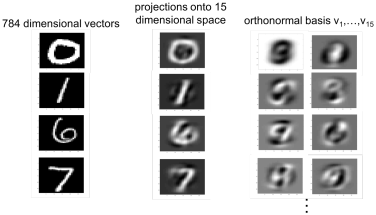
$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: orthogonal basis for subspace $\mathcal{V}$. $\mathbf{V} \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Low-Rank Approximation: Approximate $\mathbf{X} \approx \mathbf{X}\mathbf{V}\mathbf{V}^T$.



- $\mathbf{X}\mathbf{V}\mathbf{V}^T$ is a rank-k matrix – all its rows fall in $\mathcal{V}$.
- $\mathbf{X}$'s rows are approximately spanned by the columns of $\mathbf{V}$.
- $\mathbf{X}$'s columns are approximately spanned by the columns of $\mathbf{X}\mathbf{V}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: 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 $\vec{v}_1, \ldots, \vec{v}_k$.



Row (data point) compression

Column (feature) compression

10000* bathrooms+ 10* (sq. ft.) ≈ list price

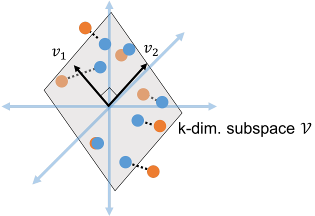
	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

BEST FIT SUBSPACE

If $\vec{x}_1, \ldots, \vec{x}_n$ are 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 $\mathbf{X}\mathbf{V}\mathbf{V}^T$. $\mathbf{X}\mathbf{V}$ gives optimal embedding of $\mathbf{X}$ in $\mathcal{V}$.

How do we find $\mathcal{V}$ (equivilantly $\mathbf{V}$)?

$$\arg\min_{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}} \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - (\mathbf{X}\mathbf{V}\mathbf{V}^T)_{i,j})^2 = \sum_{i=1}^{n} \|\vec{x}_i - \mathbf{V}\mathbf{V}^T\vec{x}_i\|_2^2 \quad \text{ar} \ldots_{\text{orthono}\ldots}$$

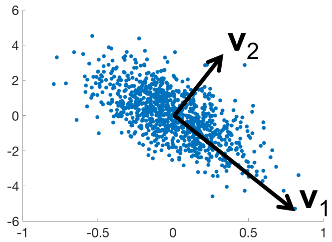


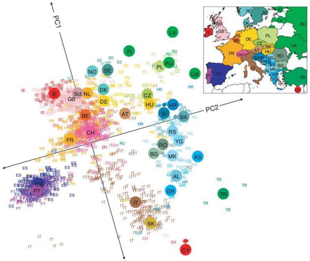
Projection only reduces data point lengths and distances. Want to minimize this reduction. How does this compare to JL random projection?

$\mathbf{V}$ minimizing $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$ is given by:

$$\underset{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}}{\arg\max} \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \sum_{i=1}^{n} \|\mathbf{V}\mathbf{V}^T \vec{x}_i\|_2^2 \quad \underset{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}}{\arg\max} \|\mathbf{X}\mathbf{V}\|_F^2 = \sum_{i=1}^{n} \|\mathbf{V}^T \vec{x}_i\|_2^2 =$$

Columns of $\mathbf{V}$ are 'directions of greatest variance' in the data.





$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: 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 $\vec{v}_1, \ldots, \vec{v}_k$.

SOLUTION VIA EIGENDECOMPOSITION

$\mathbf{V}$ minimizing $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$ is given by:

$$\underset{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}}{\arg\max} \|\mathbf{X}\mathbf{V}\|_F^2 = \sum_{i=1}^{n} \|\mathbf{V}^T \vec{x}_i\|_2^2 = \sum_{j=1}^{k} \sum_{i=1}^{n} \langle \vec{v}_j, \vec{x}_i \rangle^2 = \sum_{j=1}^{k} \|\mathbf{X}\vec{v}_j\|_2^2$$

Surprisingly, can find the columns of $\mathbf{V}$, $\vec{v}_1, \ldots, \vec{v}_k$ greedily.

$$\vec{v}_1 = \underset{\vec{v} \text{ with } \|v\|_2 = 1}{\arg\max} \|\mathbf{X}\vec{v}\|_2^2 \vec{v}^T \mathbf{X}^T \mathbf{X} \vec{v}.$$

$$\vec{v}_2 = \underset{\vec{v} \text{ with } \|v\|_2 = 1,\ \langle \vec{v}, \vec{v}_1 \rangle = 0}{\arg\max} \vec{v}^T \mathbf{X}^T \mathbf{X} \vec{v}.$$

$$\cdots$$

$$\vec{v}_k = \underset{\vec{v} \text{ with } \|v\|_2 = 1,\ \langle \vec{v}, \vec{v}_j \rangle = 0\ \forall j < k}{\arg\max} \vec{v}^T \mathbf{X}^T \mathbf{X} \vec{v}.$$

These are exactly the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: 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 $\vec{v}_1, \ldots, \vec{v}_k$.

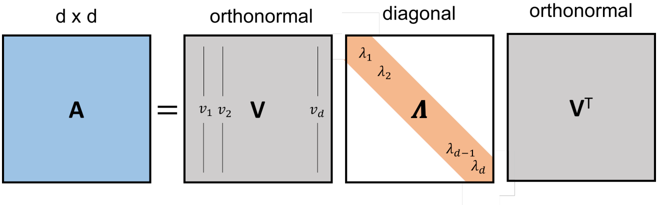
Eigenvector: $\vec{x} \in \mathbb{R}^d$ is an eigenvector of a matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ if $\mathbf{A}\vec{x} = \lambda \vec{x}$ for some scalar λ (the eigenvalue corresponding to $\vec{x}$).

- That is, $\mathbf{A}$ just 'stretches' x.
- If $\mathbf{A}$ is symmetric, can find d orthonormal eigenvectors $\vec{v}_1, \ldots, \vec{v}_d$. Let $\mathbf{V} \in \mathbb{R}^{d \times d}$ have these vectors as columns.

$$\mathbf{AV} = \begin{bmatrix} | & | & & | \\ \mathbf{A}\vec{v}_1 & \mathbf{A}\vec{v}_2 & \cdots & \mathbf{A}\vec{v}_d \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \lambda_1\vec{v}_1 & \lambda_2\vec{v}_2 & \cdots & \lambda\vec{v}_d \\ | & | & & | \end{bmatrix} = \mathbf{V\Lambda}$$

Yields eigendecomposition: $\mathbf{AVV}^T = \mathbf{A} = \mathbf{V\Lambda V}^T$.

REVIEW OF EIGENVECTORS AND EIGENDECOMPOSITION



Typically order the eigenvectors in decreasing order:
$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$.

Courant-Fischer Principal: For symmetric $\mathbf{A}$, the eigenvectors are given via the greedy optimization:

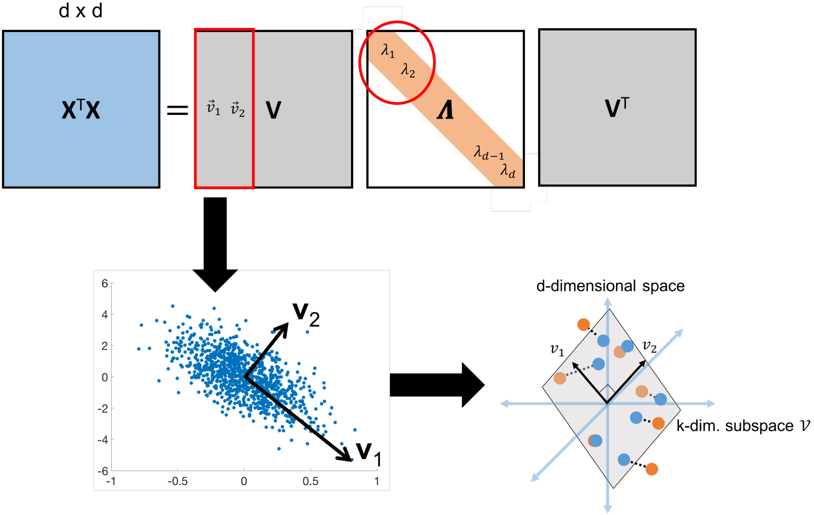
$$\vec{v}_1 = \underset{\vec{v} \text{ with } \|v\|_2=1}{\arg\max} \; \vec{v}^T \mathbf{A} \vec{v}.$$

$$\vec{v}_2 = \underset{\vec{v} \text{ with } \|v\|_2=1,\; \langle \vec{v}, \vec{v}_1 \rangle = 0}{\arg\max} \; \vec{v}^T \mathbf{A} \vec{v}.$$

$$\cdots$$

$$\vec{v}_d = \underset{\vec{v} \text{ with } \|v\|_2=1,\; \langle \vec{v}, \vec{v}_j \rangle = 0 \;\forall j < d}{\arg\max} \; \vec{v}^T \mathbf{A} \vec{v}.$$

- $\vec{v}_j^T \mathbf{A} \vec{v}_j = \lambda_j \cdot \vec{v}_j^T \vec{v}_j = \lambda_j$, the j^{th} largest eigenvalue.
- The first k eigenvectors of $\mathbf{X}^T\mathbf{X}$ (corresponding to the largest k eigenvalues) are exactly the directions of greatest variance in $\mathbf{X}$ that we use for low-rank approximation.


d x d
X^TX
=
$\vec{v}_1$ $\vec{v}_2$ V
λ_1
λ_2
Λ
λ_{d-1}
λ_d
V^T
v_2
v_1
6
4
2
0
-2
-4
-6
-1
-0.5
0
0.5
1
d-dimensional space
v_1
v_2
k-dim. subspace $\mathcal{V}$

Upshot: Letting $\mathbf{V}_k$ have columns $\vec{v}_1, \ldots, \vec{v}_k$ corresponding to the top k eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k$ is the orthogonal basis minimizing

$$\|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2,$$

This is principal component analysis (PCA).

How accurate is this low-rank approximation? Can understand using eigenvalues of $\mathbf{X}^T\mathbf{X}$.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Let $\vec{v}_1, \ldots, \vec{v}_k$ be the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$ (the top k principal components). Approximation error is:

$$\|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 \operatorname{tr}(\mathbf{X}^T\mathbf{X}) - \|\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 \operatorname{tr}(\mathbf{V}_k^T\mathbf{X}^T\mathbf{X}\mathbf{V}_k)$$

$$= \sum_{i=1}^{d} \lambda_i(\mathbf{X}^T\mathbf{X}) - \sum_{i=1}^{k} \vec{v}_i^T\mathbf{X}^T\mathbf{X}\vec{v}_i$$

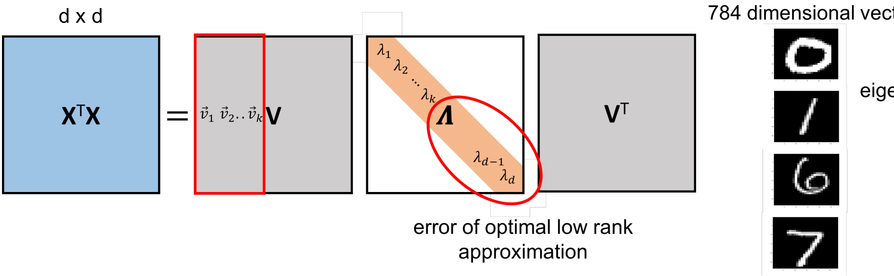
$$= \sum_{i=1}^{d} \lambda_i(\mathbf{X}^T\mathbf{X}) - \sum_{i=1}^{k} \lambda_i(\mathbf{X}^T\mathbf{X}) = \sum_{i=k+1}^{d} \lambda_i(\mathbf{X}^T\mathbf{X})$$

- For any matrix $\mathbf{A}$, $\|\mathbf{A}\|_F^2 = \sum_{i=1}^{d} \|\vec{a}_i\|_2^2 = \operatorname{tr}(\mathbf{A}^T\mathbf{A})$ (sum of diagonal entries = sum eigenvalues).

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

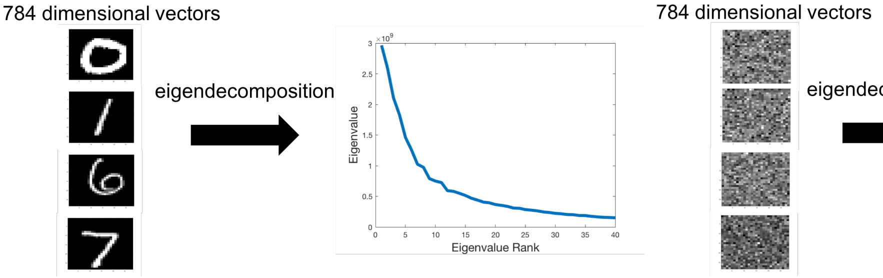
Claim: The error in approximating $\mathbf{X}$ with the best rank k approximation (projecting onto the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$ is:

$$\|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \sum_{i=k+1}^{d} \lambda_i(\mathbf{X}^T\mathbf{X})$$

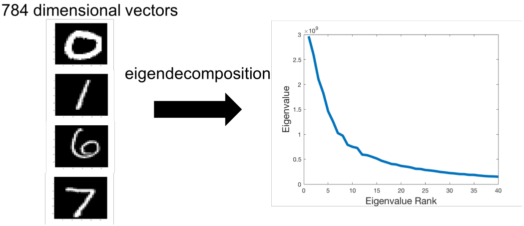


$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

Plotting the spectrum of the covariance matrix $\mathbf{X}^T\mathbf{X}$ (its eigenvalues) shows how compressible $\mathbf{X}$ is using low-rank approximation (i.e., how close $\vec{x}_1, \ldots, \vec{x}_n$ are to a low-dimensional subspace).



$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.



Exercise: Show that the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are always positive.

Hint: Use that $\lambda_j = \vec{v}_j^T \mathbf{X}^T \mathbf{X} \vec{v}_j$.

- Many (most) datasets can be approximated via projection onto a low-dimensional subspace.
- Find this subspace via a maximization problem:

$$\max_{\text{orthonormal } \mathbf{V}} \|\mathbf{X}\mathbf{V}\|_F^2.$$

- Greedy solution via eigendecomposition of $\mathbf{X}^T\mathbf{X}$.
- Columns of $\mathbf{V}$ are the top eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- Error of best low-rank approximation is determined by the tail of $\mathbf{X}^T\mathbf{X}$'s eigenvalue spectrum.

Recall: Low-rank approximation is possible when our data features are correlated.

10000* bathrooms+ 10* (sq. ft.) ≈ list price

	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

Our compressed dataset is $\mathbf{C} = \mathbf{X}\mathbf{V}_k$ where the columns of $\mathbf{V}_k$ are the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$.

What is the covariance of $\mathbf{C}$? $\mathbf{C}^T\mathbf{C} = \mathbf{V}_k^T\mathbf{X}^T\mathbf{X}\mathbf{V}_k = \mathbf{V}_k^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{V}_k = \mathbf{\Lambda}_k$

Covariance becomes diagonal. I.e., all correlations have been removed. Maximal compression.

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d \times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.

What is the runtime to compute an optimal low-rank approximation?

- Computing the covariance matrix $\mathbf{X}^T\mathbf{X}$ requires $O(nd^2)$ time.
- Computing its full eigendecomposition to obtain $\vec{v}_1, \ldots, \vec{v}_k$ requires $O(d^3)$ time (similar to the inverse $(\mathbf{X}^T\mathbf{X})^{-1}$).

Many faster iterative and randomized methods. Runtime is roughly $\tilde{O}(ndk)$ to output just to top k eigenvectors $\vec{v}_1, \ldots, \vec{v}_k$.

- Will see in a few classes

$\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$: data points, $\mathbf{X} \in \mathbb{R}^{n\times d}$: data matrix, $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^d$: top eigenvectors of $\mathbf{X}^T\mathbf{X}$, $\mathbf{V}_k \in \mathbb{R}^{d\times k}$: matrix with columns $\vec{v}_1, \ldots, \vec{v}_k$.