Dimensionality Reduction

CS771: Introduction to Machine Learning

Dimensionality Reduction

- Goal: Reduce the dimensionality of each input $\boldsymbol{x}_n \in \mathbb{R}^D$

$\boldsymbol{z}_n \in \mathbb{R}^K$ $(K \ll D)$ is a compressed version of $\boldsymbol{x}_n$

$$\boldsymbol{z}_n = f(\boldsymbol{x}_n)$$

- Also want to be able to (approximately) reconstruct $\boldsymbol{x}_n$ from $\boldsymbol{z}_n$

$$\tilde{\boldsymbol{x}}_n = g(\boldsymbol{z}_n) = g(f(\boldsymbol{x}_n)) \approx \boldsymbol{x}_n$$

- Sometimes f is called "encoder" and g is called "decoder". Can be linear/nonlinear

- These functions are learned by minimizing the distortion/reconstruction error of inputs

$$\mathcal{L} = \sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \sum_{n=1}^{N} \|\boldsymbol{x}_n - g(f(\boldsymbol{x}_n))\|^2$$

Image source: Manjon et al (2013): Diffusion Weighted Image Denoising using overcomplete Local PCA

Dimensionality Reduction

- Choosing f and g as linear transformations $\boldsymbol{W}^T$ $(K \times D)$ and $\boldsymbol{W}$, respectively

$\boldsymbol{z}_n$

$\tilde{\boldsymbol{x}}_n = \boldsymbol{W}\boldsymbol{z}_n$

$$\mathcal{L} = \sum_{n=1}^{N} ||\boldsymbol{x}_n - g(f(\boldsymbol{x}_n))||^2 = \sum_{n=1}^{N} ||\boldsymbol{x}_n - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}_n||^2$$

Principal Component Analysis (PCA). Proof shortly

- Minimizer of $\mathcal{L}$, if the K columns of $\boldsymbol{W}$ are orthonormal, are top K eigenvectors of

$D \times D$ empirical covariance matrix of inputs

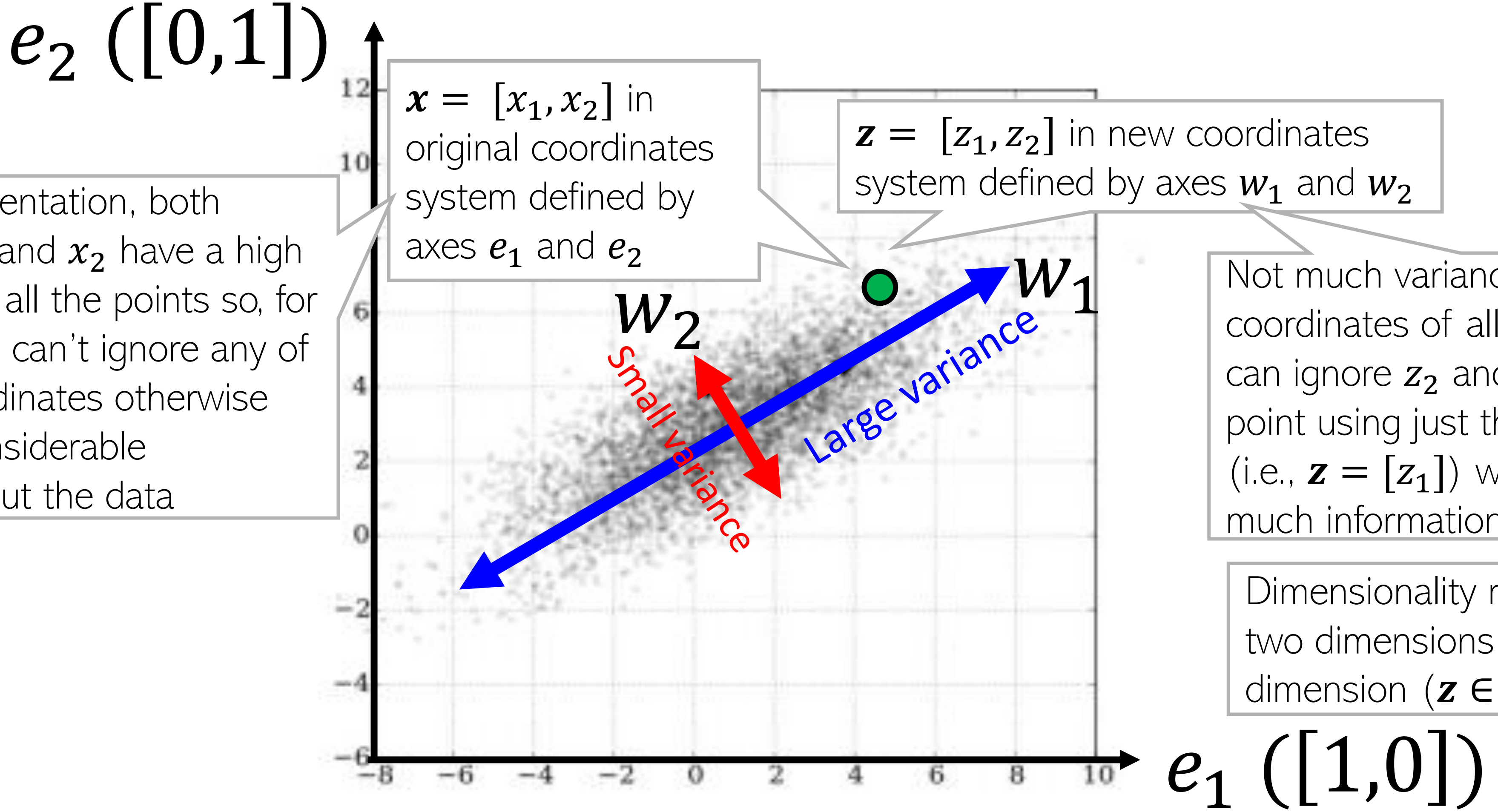
$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^\top = \frac{1}{N}\boldsymbol{X}_c^\top \boldsymbol{X}_c$$

$\boldsymbol{X}_c$ is the $N \times D$ matrix of inputs after centering each input (subtracting off the mean of inputs from each input)

- The matrix $\boldsymbol{W}$ does a "linear projection" of each input $\boldsymbol{x}_n \in \mathbb{R}^D$ into a K dim space
  - $\boldsymbol{z}_n = \boldsymbol{W}^T\boldsymbol{x}_n \in \mathbb{R}^K$ denotes this linear projection
- Note: If we use $K = D$ eigenvectors for $\boldsymbol{W}$, the reconstruction will be perfect $(\mathcal{L} = 0)$

Principal Component Analysis (PCA)

- PCA learns a different and more economical coordinate system to represent data



$\boldsymbol{x} = [x_1, x_2]$ in original coordinates system defined by axes $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$

$\boldsymbol{z} = [z_1, z_2]$ in new coordinates system defined by axes $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$

With this representation, both coordinates x_1 and x_2 have a high variance across all the points so, for any point $\boldsymbol{x}$, we can't ignore any of these two coordinates otherwise we will lose considerable information about the data

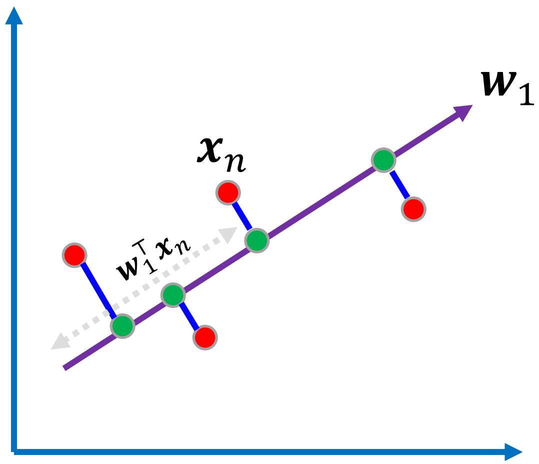
Not much variance in the z_2 coordinates of all the points so we can ignore z_2 and represent each point using just the first coordinate (i.e., $\boldsymbol{z} = [z_1]$) without losing much information about the data

Dimensionality reduction! From two dimensions ($\boldsymbol{x} \in \mathbb{R}^2$) to one dimension ($\boldsymbol{z} \in \mathbb{R}$)

- Top eigenvectors of the covariance matrix of inputs give us the large variance directions

Finding Max. Variance Directions

- Consider projecting an input $\boldsymbol{x}_n \in \mathbb{R}^D$ along a direction $\boldsymbol{w}_1 \in \mathbb{R}^D$
- Projection/embedding of $\boldsymbol{x}_n$ (red points below) will be $\boldsymbol{w}_1^\top \boldsymbol{x}_n$ (green pts below)



Mean of projections of all inputs:

$$\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{w}_1^\top \boldsymbol{x}_n = \boldsymbol{w}_1^\top \left(\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n\right) = \boldsymbol{w}_1^\top \boldsymbol{\mu}$$

$\boldsymbol{S}$ is the $D \times D$ cov matrix of the data:

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^\top$$

Variance of the projections:

$$\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{w}_1^\top \boldsymbol{x}_n - \boldsymbol{w}_1^\top \boldsymbol{\mu})^2 = \frac{1}{N}\sum_{n=1}^{N} \{\boldsymbol{w}_1^\top (\boldsymbol{x}_n - \boldsymbol{\mu})\}^2 = \boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$$

For already centered data, $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top$

- Want $\boldsymbol{w}_1$ such that variance $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$ is maximized

$$\underset{\boldsymbol{w}_1}{\text{argmax}}\ \boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 \quad \text{s.t.} \quad \boldsymbol{w}_1^\top \boldsymbol{w}_1 = 1$$

Need this constraint otherwise the objective's max will be infinity

Max. Variance Direction

Variance along the direction $\boldsymbol{w}_1$

Note: Total variance of the data is equal to the sum of eigenvalues of $\boldsymbol{S}$, i.e., $\sum_{d=1}^{D} \lambda_d$

PCA would keep the top $K < D$ such directions of largest variances

- Our objective function was $\underset{\boldsymbol{w}_1}{\operatorname{argmax}}\, \boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$ s.t. $\boldsymbol{w}_1^\top \boldsymbol{w}_1 = 1$
- Can construct a Lagrangian for this problem

$$\underset{\boldsymbol{w}_1}{\operatorname{argmax}}\, \boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 + \lambda_1 (1\text{-}\boldsymbol{w}_1^\top \boldsymbol{w}_1)$$

- Taking derivative w.r.t. $\boldsymbol{w}_1$ and setting to zero gives $\boldsymbol{S}\boldsymbol{w}_1 = \lambda_1 \boldsymbol{w}_1$

Note: In general, $\boldsymbol{S}$ will have D eigvecs

- Therefore $\boldsymbol{w}_1$ is an eigenvector of the cov matrix $\boldsymbol{S}$ with eigenvalue λ_1
- Claim: $\boldsymbol{w}_1$ is the eigenvector of $\boldsymbol{S}$ with largest eigenvalue λ_1. Note that

$$\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1 = \lambda_1 \boldsymbol{w}_1^\top \boldsymbol{w}_1 = \lambda_1$$

- Thus variance $\boldsymbol{w}_1^\top \boldsymbol{S} \boldsymbol{w}_1$ will be max. if λ_1 is the largest eigenvalue (and $\boldsymbol{w}_1$ is the corresponding top eigenvector; also known as the first Principal Component)
- Other large variance directions can also be found likewise (with each being orthogonal to all others) using the eigendecomposition of cov matrix $\boldsymbol{S}$ (this is PCA)

The PCA Algorithm

- Center the data (subtract the mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$ from each data point)
- Compute the $D \times D$ covariance matrix $\mathbf{S}$ using the centered data matrix $\mathbf{X}$ as

$$\mathbf{S} = \frac{1}{N}\mathbf{X}^\top\mathbf{X}$$

(Assuming $\mathbf{X}$ is arranged as $N \times D$)

- Do an eigendecomposition of the covariance matrix $\mathbf{S}$ (many methods exist)
- Take top $K < D$ leading eigvectors $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K\}$ with eigvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$
- The K-dimensional projection/embedding of each input is

$$\boldsymbol{z}_n \approx \mathbf{W}_K^\top \boldsymbol{x}_n$$

$\mathbf{W}_K = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$ is the "projection matrix" of size $D \times K$

Note: Can decide how many eigvecs to use based on how much variance we want to capture (recall that each λ_k gives the variance in the k^{th} direction (and their sum is the total variance)

The Reconstruction Error View of PCA

- Representing a data point $\boldsymbol{x}_n = [x_{n1}, x_{n2}, \ldots, x_{nD}]^\top$ in the standard orthonormal basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_D\}$

x_{nd} is the coordinate of $\boldsymbol{x}_n$ along the direction $\boldsymbol{e}_d$

$$\boldsymbol{x}_n = \sum_{d=1}^{D} x_{nd}\boldsymbol{e}_d$$

$\boldsymbol{e}_d$ is a vector of all zeros except a single 1 at the d^{th} position. Also, $\boldsymbol{e}_d^\top \boldsymbol{e}_{d'} = 0$ for $d \neq d'$

- Let's represent the same data point in a new orthonormal basis $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$

z_{nd} is the projection/coordinate of $\boldsymbol{x}_n$ along the direction $\boldsymbol{w}_d$ since $z_{nd} = \boldsymbol{w}_d^\top \boldsymbol{x}_n = \boldsymbol{x}_n^\top \boldsymbol{w}_d$ (verify)

$$\boldsymbol{x}_n = \sum_{d=1}^{D} z_{nd}\boldsymbol{w}_d$$

$\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nD}]^\top$ denotes the co-ordinates of $\boldsymbol{x}_n$ in the new basis

- Ignoring directions along which projection z_{nd} is small, we can approximate $\boldsymbol{x}_n$ as

$$\boldsymbol{x}_n \approx \hat{\boldsymbol{x}}_n = \sum_{d=1}^{K} z_{nd}\boldsymbol{w}_d = \sum_{d=1}^{K} (\boldsymbol{x}_n^\top \boldsymbol{w}_d)\boldsymbol{w}_d = \sum_{d=1}^{K} (\boldsymbol{w}_d \boldsymbol{w}_d^\top)\boldsymbol{x}_n$$

Note that $\left\|\boldsymbol{x}_n - \sum_{d=1}^{K}(\boldsymbol{w}_d \boldsymbol{w}_d^\top)\boldsymbol{x}_n\right\|^2$ is the reconstruction error on $\boldsymbol{x}_n$. Would like it to minimize w.r.t. $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$

- Now $\boldsymbol{x}_n$ is represented by $K < D$ dim. rep. $\boldsymbol{z}_n = [z_{n1}, z_{n2}, \ldots, z_{nK}]$ and

Also, $\boldsymbol{x}_n \approx \mathbf{W}_K \boldsymbol{z}_n$

$$\boldsymbol{z}_n = \boldsymbol{W}_K^\top \boldsymbol{x}_n$$

$\boldsymbol{W}_K = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$ is the "projection matrix" of size $D \times K$

PCA Minimizes Reconstruction Error

- We plan to use only K directions $[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$ so would like them to be such that the total reconstruction error is minimized

$$\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K) = \sum_{n=1}^{N} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\|^2 = \sum_{n=1}^{N} \left\| \boldsymbol{x}_n - \sum_{d=1}^{K} (\boldsymbol{w}_d \boldsymbol{w}_d^\top) \boldsymbol{x}_n \right\|^2$$

Constant; doesn't depend on the $\boldsymbol{w}_d$'s

Variance along $\boldsymbol{w}_d$

$$= C - \sum_{d=1}^{K} \boldsymbol{w}_d^\top \mathbf{S} \boldsymbol{w}_d \text{ (verify)}$$

- Each optimal $\boldsymbol{w}_d$ can be found by solving

$$\underset{\boldsymbol{w}_d}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K) = \underset{\boldsymbol{w}_d}{\operatorname{argmax}}\ \boldsymbol{w}_d^\top \mathbf{S} \boldsymbol{w}_d$$

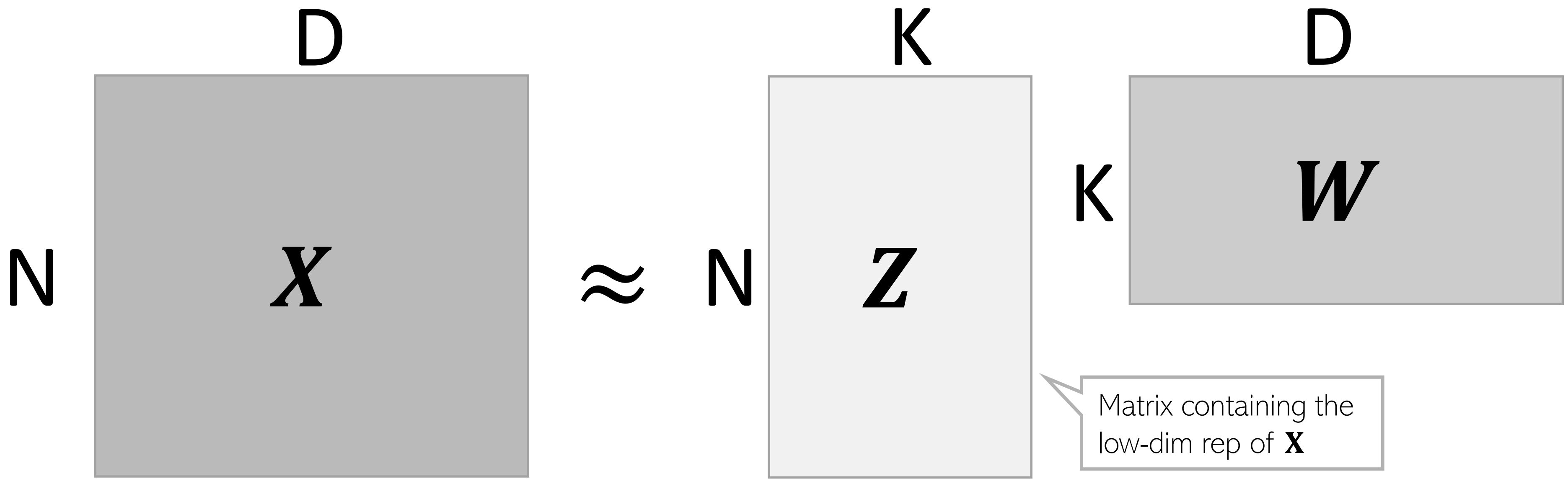
Subject to $\boldsymbol{w}_d^\top \boldsymbol{w}_d = 1$

- Thus minimizing the reconstruction error is equivalent to maximizing variance
- The K directions can be found by solving the eigendecomposition of $\mathbf{S}$

Dim-Red as Matrix Factorization

In PCA, we do want these constraints

- If we don't care about the orthonormality constraints on $\boldsymbol{W}$, then dim-red can also be achieved by solving a matrix factorization problem on the data matrix $\mathbf{X}$



Matrix containing the low-dim rep of $\mathbf{X}$

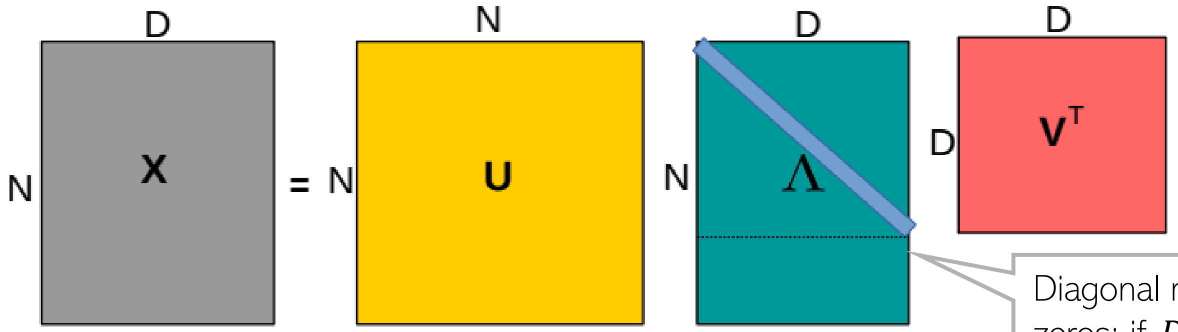
$$\{\hat{\boldsymbol{Z}}, \widehat{\boldsymbol{W}}\} = \text{argmin}_{\boldsymbol{Z},\boldsymbol{W}} ||\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}||^2$$

If $K < \min\{D, N\}$, such a factorization gives a low-rank approximation of the data matrix $\boldsymbol{X}$

- Can solve such problems using ALT-OPT
- Can impose various constraints on $\mathbf{Z}$ and $\mathbf{W}$ (e.g., sparsity, non-negativity, etc)

Singular Value Decomposition (SVD)

- Any matrix $\mathbf{X}$ of size $N \times D$ can be represented as the following decomposition



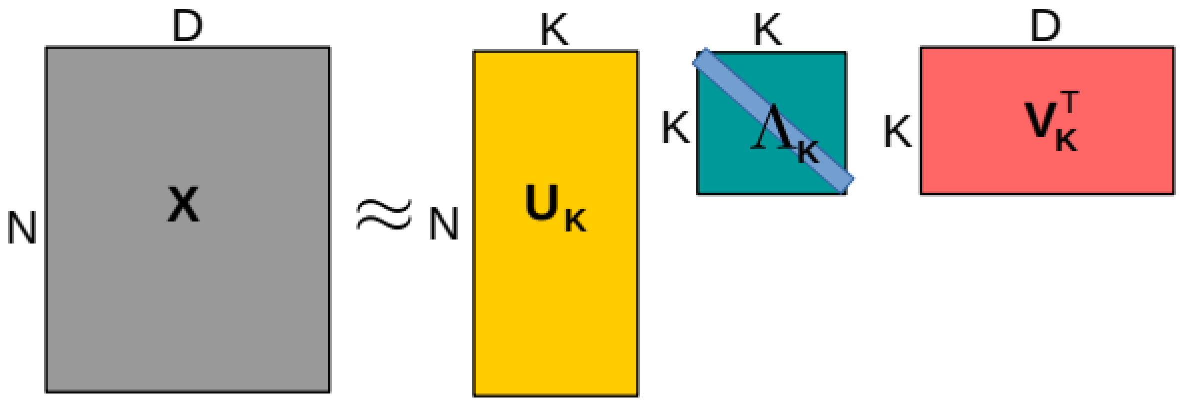
$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top = \sum_{k=1}^{\min\{N,D\}} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top$$

Diagonal matrix. If $N > D$, last $D - N$ rows are all zeros; if $D > N$, last $D - N$ columns are all zeros

- $\mathbf{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]$ is $N \times N$ matrix of left singular vectors, each $\boldsymbol{u}_n \in \mathbb{R}^N$
  - $\mathbf{U}$ is also orthonormal ($\boldsymbol{u}_n^\top \boldsymbol{u}_n = 1 \, \forall n$ and $\boldsymbol{u}_n^\top \boldsymbol{u}_{n'} = 0 \, \forall n \neq n'$)
- $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]$ is $D \times D$ matrix of right singular vectors, each $\boldsymbol{v}_d \in \mathbb{R}^D$
  - $\mathbf{V}$ is also orthonormal($\boldsymbol{v}_d^\top \boldsymbol{v}_d = 1 \, \forall d$ and $\boldsymbol{v}_d^\top \boldsymbol{v}_{d'} = 0 \, \forall d \neq d'$)
- Λ is $N \times D$ with only $\mathbf{min}(N, D)$ diagonal entries - singular values
- Note: If $\mathbf{X}$ is symmetric then it is known as eigenvalue decomposition ($\mathbf{U} = \mathbf{V}$)

Low-Rank Approximation via SVD

- If we just use the top $K < \min\{N, D\}$ singular values, we get a rank-K SVD



This is a rank-1 matrix

$$\mathbf{X} \approx \hat{\mathbf{X}} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top = \mathbf{U}_K \Lambda_K \mathbf{V}_K^\top$$

- Above SVD approx. can be shown to minimize the reconstruction error $\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|$
  - Fact: SVD gives the best rank-K approximation of a matrix

- PCA is done by doing SVD on the covariance matrix $\boldsymbol{S}$ (left and right singular vectors are the same and become eigenvectors, singular values become eigenvalues)