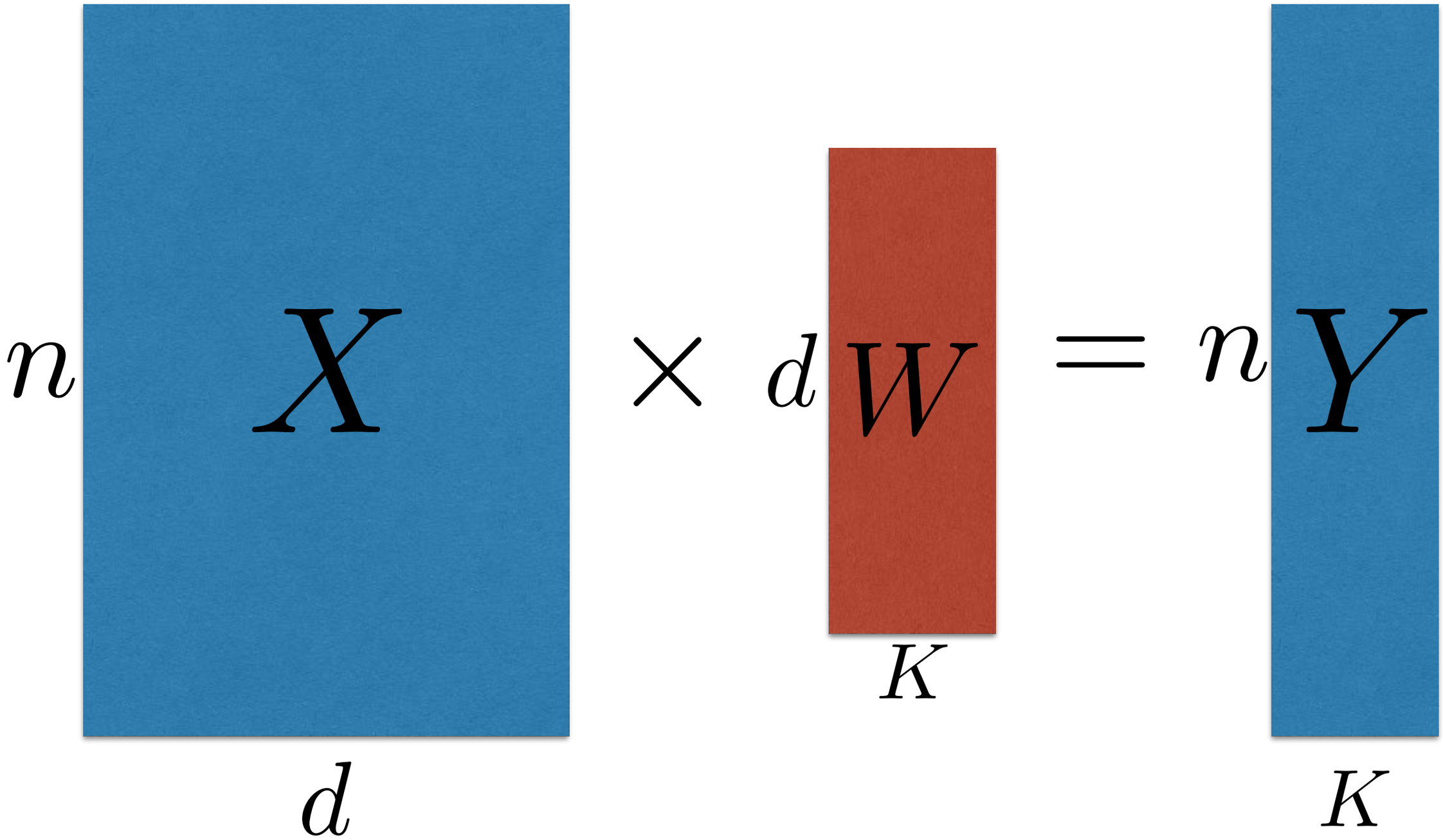
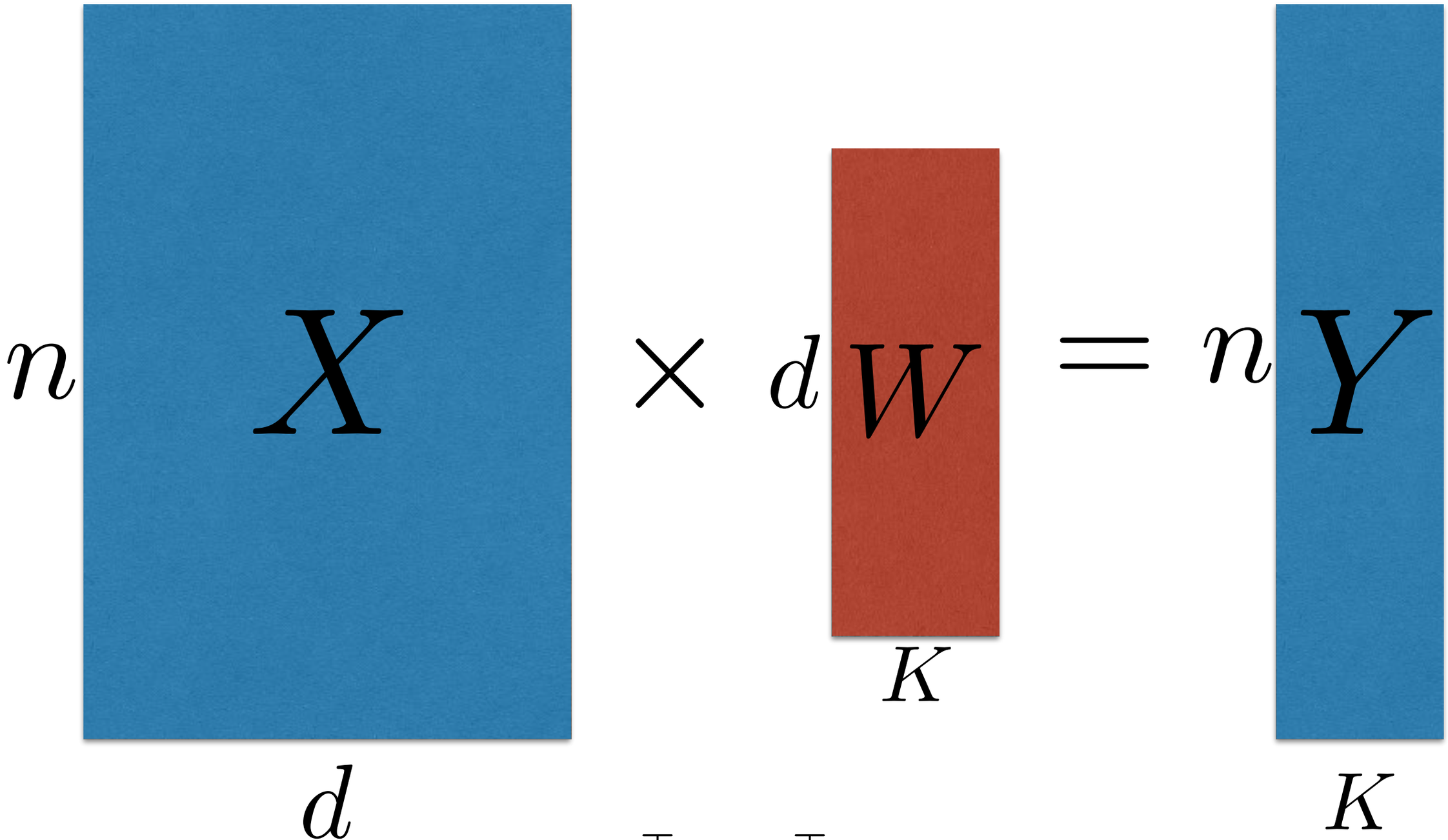
Machine Learning for Data Science (CS4786)
Lecture 4

PCA and Random Projections

Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation



$$\mathbf{y}_i^\top = \mathbf{x}_i^\top W$$

PCA: Variance Maximization

- Pick directions along which data varies the most

$$\text{Spread} = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right) = \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

PCA: Variance Maximization

- Pick directions along which data varies the most

$$\text{Spread} = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right) = \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{Maximize} \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

PCA: Variance Maximization

- Pick directions along which data varies the most

$$\text{Spread} = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right) = \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{Maximize} \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

Σ is the covariance matrix

PCA: Variance Maximization

- Pick directions along which data varies the most

$$\text{Spread} = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right) = \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{Maximize} \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

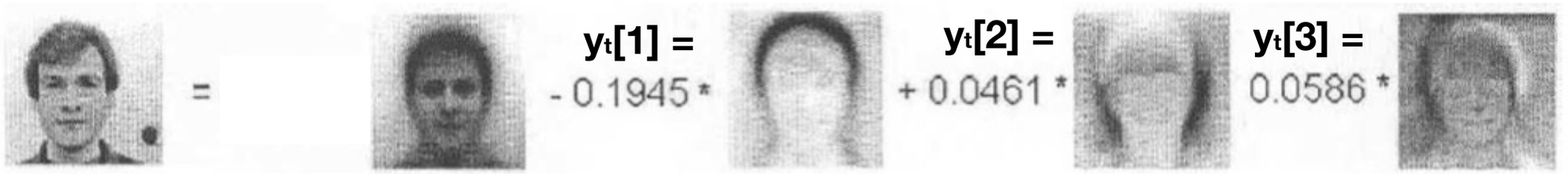
Σ is the covariance matrix

This solutions is given by W = Top K eigenvectors of Σ

Principal Component Analysis (PCA)

Turk & Pentland'91

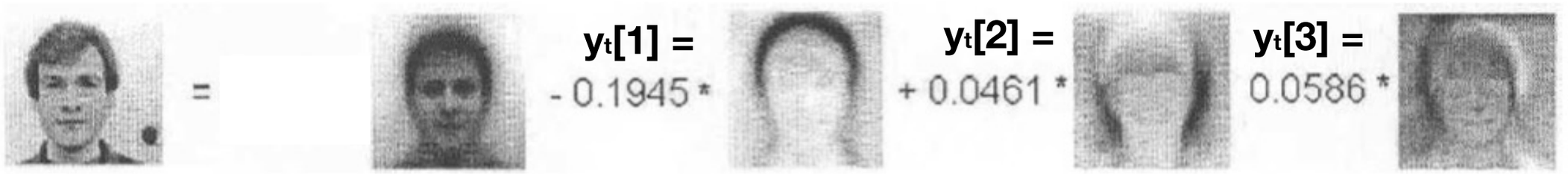
Eigen Face:



Principal Component Analysis (PCA)

Turk & Pentland'91

Eigen Face:

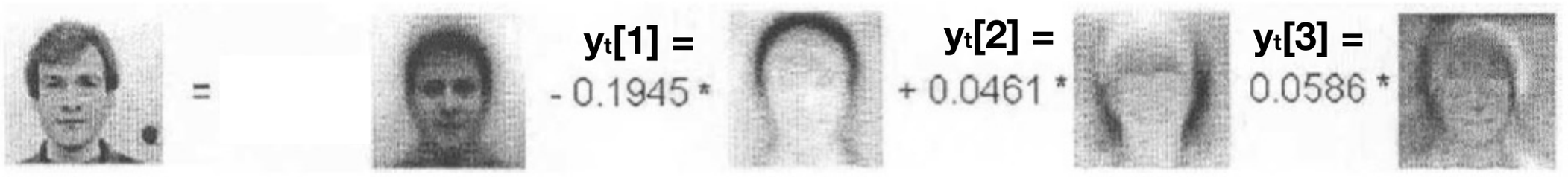


- Each x_t (each row of X) is a face image (vectorized version)

Principal Component Analysis (PCA)

Turk & Pentland'91

Eigen Face:



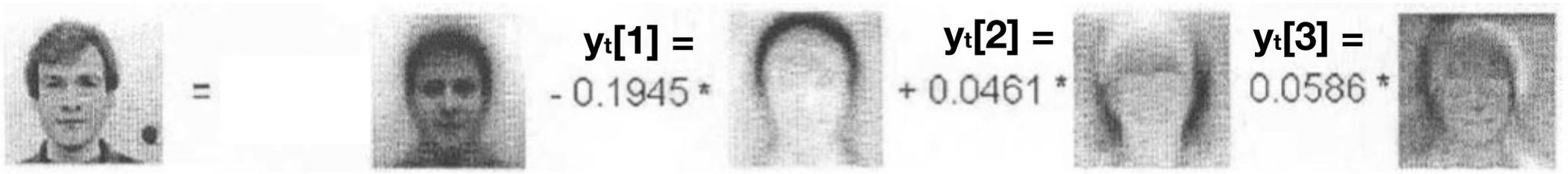
- Each x_t (each row of X) is a face image (vectorized version)

- Each y_t is the set of coefficients we multiply to the eigen face

Principal Component Analysis (PCA)

Turk & Pentland'91

Eigen Face:



- Each x_t (each row of X) is a face image (vectorized version)
- Each y_t is the set of coefficients we multiply to the eigen face
- w_i's are orthogonal to each other and of unit length

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,
- (Centered) Data-points as linear combination of some orthonormal basis, i.e.

$$\mathbf{x}_t = \mu + \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j$$

where $\mathbf{w}_1, \ldots, \mathbf{w}_d \in \mathbb{R}^d$ are the orthonormal basis and $\mu = \frac{1}{n}\sum_{t=1}^{n} x_t$.

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,
- (Centered) Data-points as linear combination of some orthonormal basis, i.e.

$$\mathbf{x}_t = \mu + \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j$$

where $\mathbf{w}_1, \ldots, \mathbf{w}_d \in \mathbb{R}^d$ are the orthonormal basis and $\mu = \frac{1}{n}\sum_{t=1}^{n} x_t$.

- Represent data as linear combination of just K orthonormal basis,

$$\hat{\mathbf{x}}_t = \mu + \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

$$= \left\| \sum_{j=K+1}^{d} \mathbf{y}_t[j]\mathbf{w}_j \right\|_2^2$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

$$= \left\| \sum_{j=K+1}^{d} \mathbf{y}_t[j]\mathbf{w}_j \right\|_2^2 \quad (\text{but } \|a+b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 + 2a^\top b)$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

$$= \left\| \sum_{j=K+1}^{d} \mathbf{y}_t[j]\mathbf{w}_j \right\|_2^2 \quad (\text{but } \|a+b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 + 2a^\top b)$$

$$= \left(\sum_{j=K+1}^{d} \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 + 2 \sum_{j=K+1}^{d} \sum_{i=j+1}^{d} \mathbf{y}_t[j]\mathbf{y}_t[i]\mathbf{w}_j^\top \mathbf{w}_i \right)$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

$$= \left\| \sum_{j=K+1}^{d} \mathbf{y}_t[j]\mathbf{w}_j \right\|_2^2 \quad (\text{but } \|a+b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 + 2a^\top b)$$

$$= \left(\sum_{j=K+1}^{d} \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 + 2 \sum_{j=K+1}^{d} \sum_{i=j+1}^{d} \mathbf{y}_t[j]\mathbf{y}_t[i]\mathbf{w}_j^\top \mathbf{w}_i \right)$$

$$= \sum_{j=k+1}^{d} \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2$$

PCA: Minimizing Reconstruction Error

- Goal: find the basis that minimizes reconstruction error,

$$\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2$$

$$= \left\| \sum_{j=1}^{K} \mathbf{y}_t[j]\mathbf{w}_j + \mu - \sum_{j=1}^{d} \mathbf{y}_t[j]\mathbf{w}_j - \mu \right\|_2^2$$

$$= \left\| \sum_{j=K+1}^{d} \mathbf{y}_t[j]\mathbf{w}_j \right\|_2^2 \quad (\text{but } \|a+b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 + 2a^\top b)$$

$$= \left(\sum_{j=K+1}^{d} \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 + 2 \sum_{j=K+1}^{d} \sum_{i=j+1}^{d} \mathbf{y}_t[j]\mathbf{y}_t[i]\mathbf{w}_j^\top \mathbf{w}_i \right)$$

$$= \sum_{j=k+1}^{d} \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 \quad (\text{last step because } \mathbf{w}_j \perp \mathbf{w}_i)$$

PCA: Minimizing Reconstruction Error

$$\frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2\|\mathbf{w}_j\|_2^2 \qquad (\text{but } \|\mathbf{w}_j\| = 1)$$

PCA: Minimizing Reconstruction Error

$$\frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2\|\mathbf{w}_j\|_2^2 \qquad (\text{but } \|\mathbf{w}_j\| = 1)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2$$

PCA: Minimizing Reconstruction Error

$$\frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2\|\mathbf{w}_j\|_2^2 \quad (\text{but } \|\mathbf{w}_j\| = 1)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2 \quad (\text{now } \mathbf{y}_j = \mathbf{w}_j^\top(\mathbf{x}_t - \mu))$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}(\mathbf{w}_j^\top(\mathbf{x}_t - \mu))^2$$

PCA: Minimizing Reconstruction Error

$$\frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2\|\mathbf{w}_j\|_2^2 \quad (\text{but } \|\mathbf{w}_j\| = 1)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2 \quad (\text{now } \mathbf{y}_j = \mathbf{w}_j^\top(\mathbf{x}_t - \mu))$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}(\mathbf{w}_j^\top(\mathbf{x}_t - \mu))^2$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{w}_j^\top(\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top\mathbf{w}_j$$

PCA: Minimizing Reconstruction Error

$$\frac{1}{n}\sum_{t=1}^{n}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2\|\mathbf{w}_j\|_2^2 \quad (\text{but } \|\mathbf{w}_j\| = 1)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{y}_t[j]^2 \quad (\text{now } \mathbf{y}_j = \mathbf{w}_j^\top(\mathbf{x}_t - \mu))$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}(\mathbf{w}_j^\top(\mathbf{x}_t - \mu))^2$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{j=k+1}^{d}\mathbf{w}_j^\top(\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top\mathbf{w}_j$$

$$= \sum_{j=k+1}^{d}\mathbf{w}_j^\top\Sigma\mathbf{w}_j$$

PCA: Minimizing Reconstruction Error

PCA: Minimizing Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \;\&\; \mathbf{w}_j \perp \mathbf{w}_i$$

Maximize Total Spread

$$\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \mathrm{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)$$

Maximize Total Spread

$$\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \mathrm{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)$$

Minimize Reconstruction Error

$$\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2$$

Maximize Total Spread

$$\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)$$

Minimize Reconstruction Error

$$\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2$$

$$\Updownarrow$$

$$\text{Maximize } \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

Maximize Total Spread

$$\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \text{dist}^2 \left(y_t[j], \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)$$

$$\Updownarrow$$

$$\text{Maximize} \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

Minimize Reconstruction Error

$$\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2$$

$$\Updownarrow$$

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \ \& \ \mathbf{w}_j \perp \mathbf{w}_i$$

Claim

Claim

Maximize Total Spread = Minimize Reconstruction Error

PCA: Minimizing Reconstruction Error

$$\text{Claim: } \sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j = \text{ Constant } = \frac{1}{n} \sum_{t=1}^{n} \|x_t - \mu\|_2^2$$

PCA: Minimizing Reconstruction Error

$$\text{Claim: } \sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j = \text{ Constant } = \frac{1}{n} \sum_{t=1}^{n} \|x_t - \mu\|_2^2$$

$$\text{Recall that: } \frac{1}{n} \sum_{t=1}^{n} \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

PCA: Minimizing Reconstruction Error

$$\text{Claim: } \sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j = \text{ Constant } = \frac{1}{n} \sum_{t=1}^{n} \|x_t - \mu\|_2^2$$

$$\text{Recall that: } \frac{1}{n} \sum_{t=1}^{n} \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{Take K} = 0 \text{ so that } \hat{\mathbf{x}}_t = \mu$$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \mu\|_2^2 - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \mu\|_2^2 - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(- \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\sum_{j=1}^{d} \mathbf{w}_j^\top \Sigma \mathbf{w}_j - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \mu\|_2^2 - \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(- \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Maximize} \sum_{j=1}^{K} \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

Principal Component Analysis

1. $$\Sigma = \mathrm{cov}\left(X\right)$$

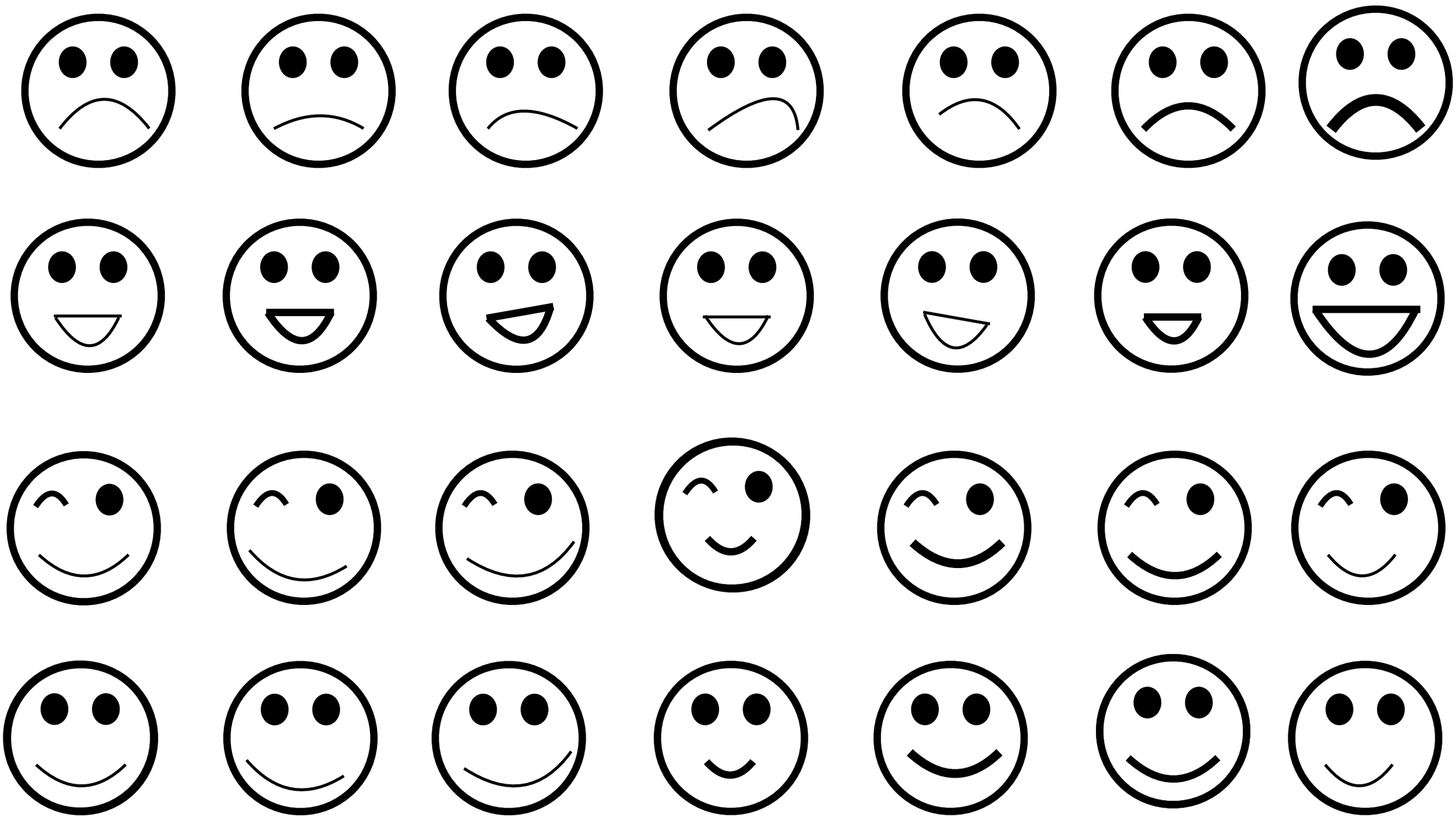
2. $$W = \mathrm{eigs}\left(\Sigma, K\right)$$

3. $$Y = X - \mu \times W$$

Reconstruction

4. $\hat{X} = Y \times W^{\top} + \mu$

Principal Component Analysis: Demo



When $d >> n$

- If $d >> n$ then Σ is large

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^\top$$

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^\top$$

$$V^\top V = I$$
$$U^\top U = I$$

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^\top$$

$$V^\top V = I$$
$$U^\top U = I$$

Then note that, $\Sigma = (X - \mu)^\top (X - \mu) = VD^2V$

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^\top$$

$$V^\top V = I$$
$$U^\top U = I$$

Then note that, $\Sigma = (X - \mu)^\top (X - \mu) = VD^2V$

- Hence, matrix V is the same as matrix W got from eigen decomposition of Σ, eigenvalues are diagonal elements of D^2

When $d >> n$

- If $d >> n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^T$$

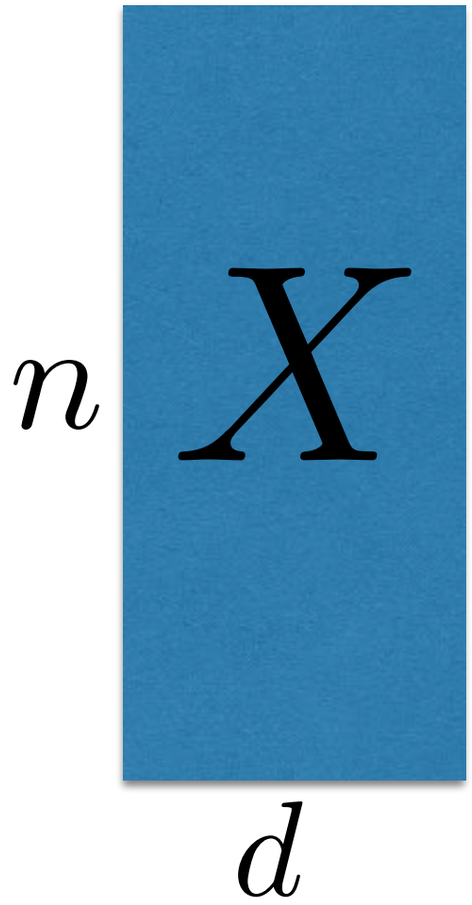
$$V^T V = I$$
$$U^T U = I$$

Then note that, $\Sigma = (X - \mu)^T (X - \mu) = VD^2V$

- Hence, matrix V is the same as matrix W got from eigen decomposition of Σ, eigenvalues are diagonal elements of D^2
- Alternative algorithm:

$$[U, V] = \text{SVD}(X - \mu, K) \quad W = V$$

The Tall, the Fat and the Ugly



The Tall, the Fat and the Ugly

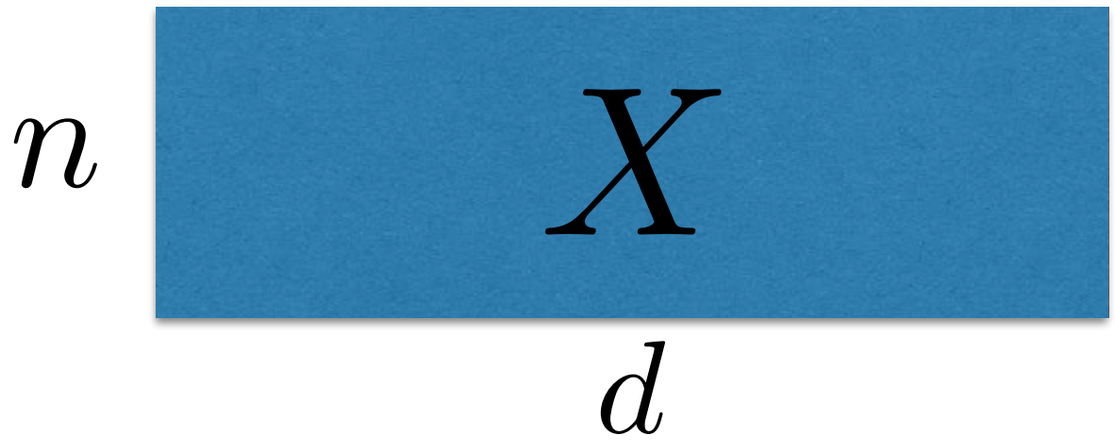
$$\underset{d}{\boxed{X^\top}}\underset{n}{\boxed{}} \times \underset{n}{\boxed{X}}\underset{d}{\boxed{}} \Big/ n = \underset{d}{\boxed{\Sigma}}\underset{d}{\boxed{}}$$

The Tall, the Fat and the Ugly

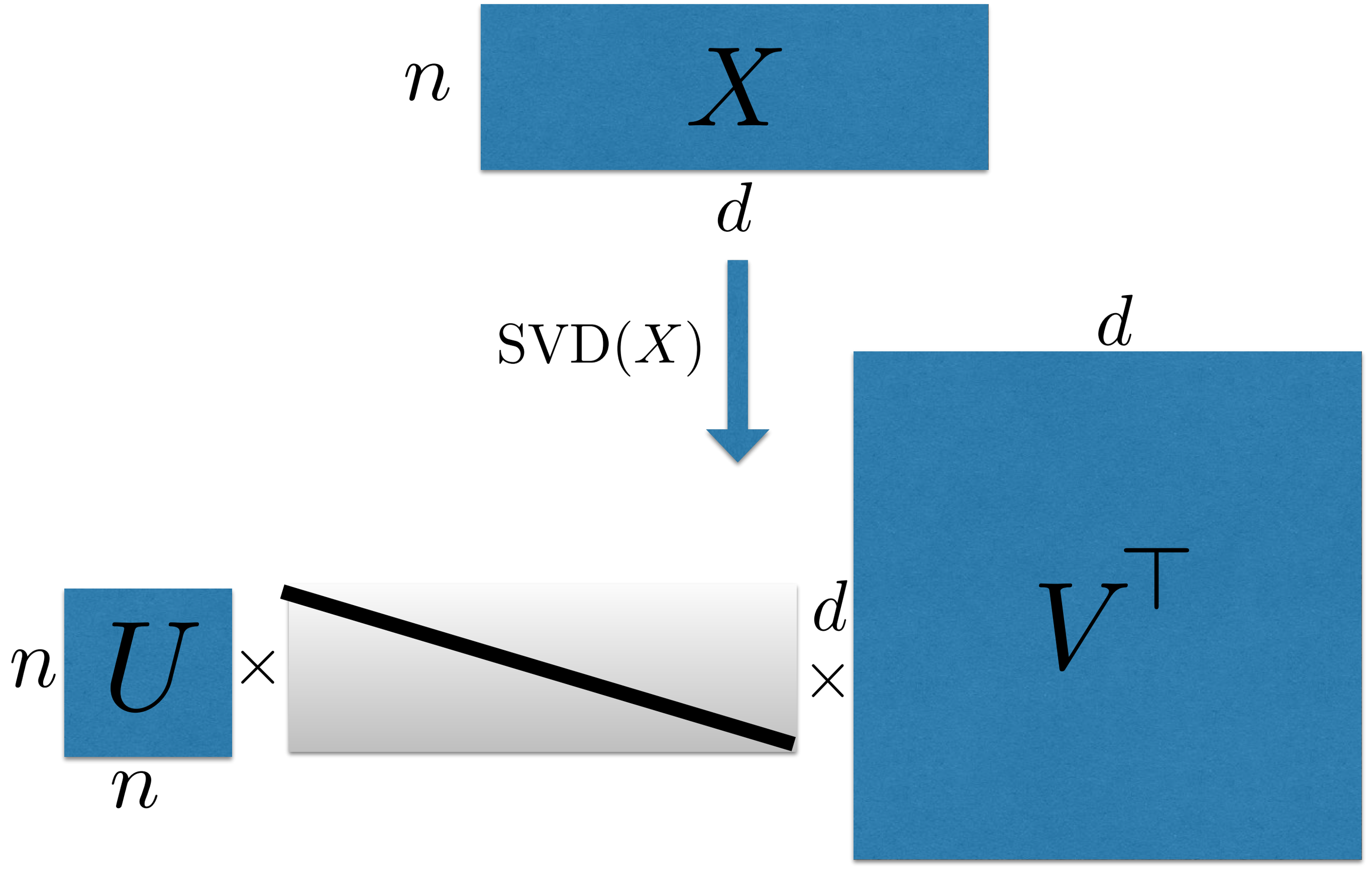
$$\underset{d}{\square}\underset{n}{\square^\top} \times \underset{n}{\square}\underset{d}{\square} \Big/ n = \underset{d}{\square}$$

$$\underset{d}{\square}\underset{K}{\square} = \text{Eigs}\left(\square, K\right)$$

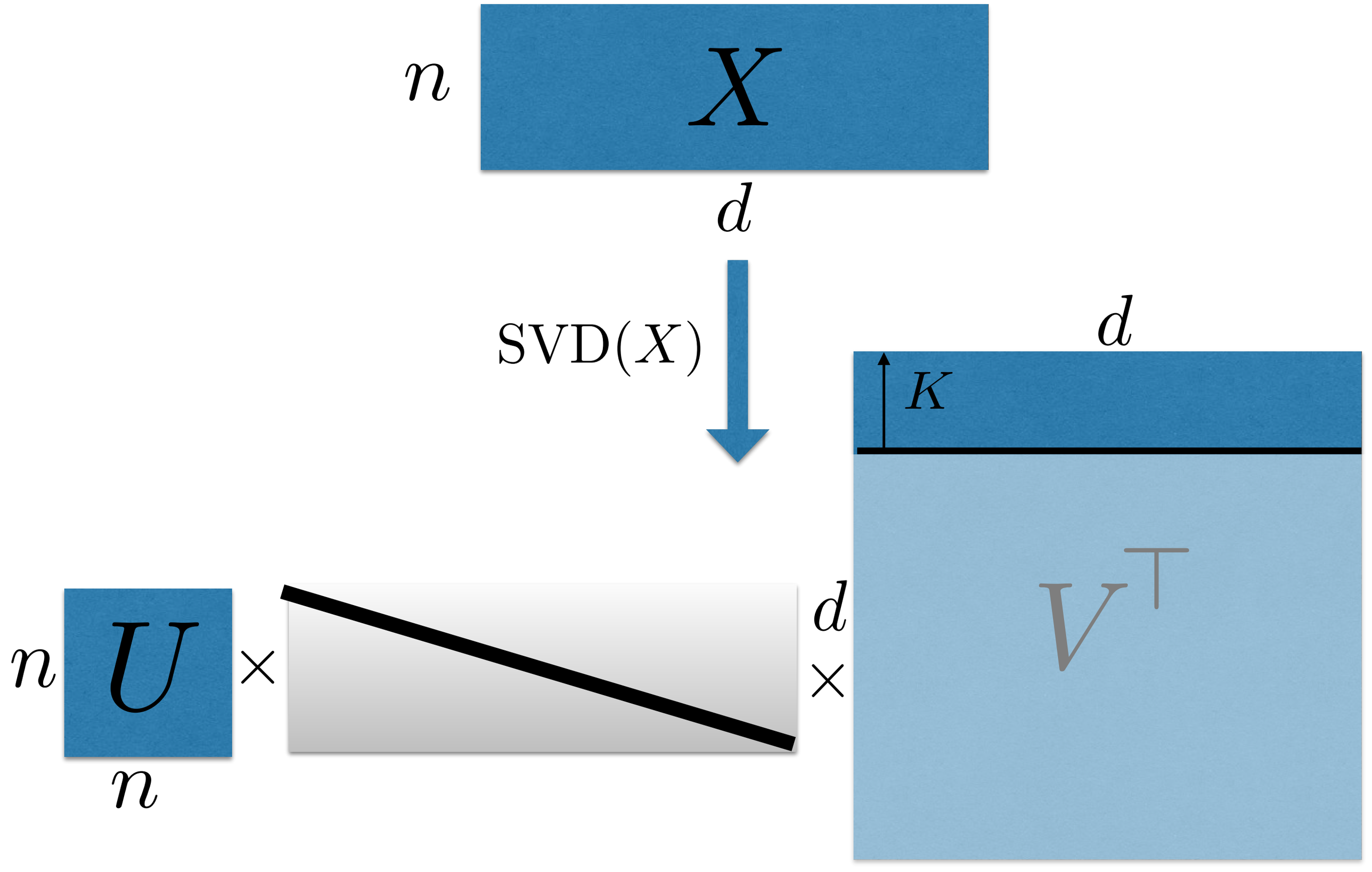
The Tall, **the Fat** and the Ugly



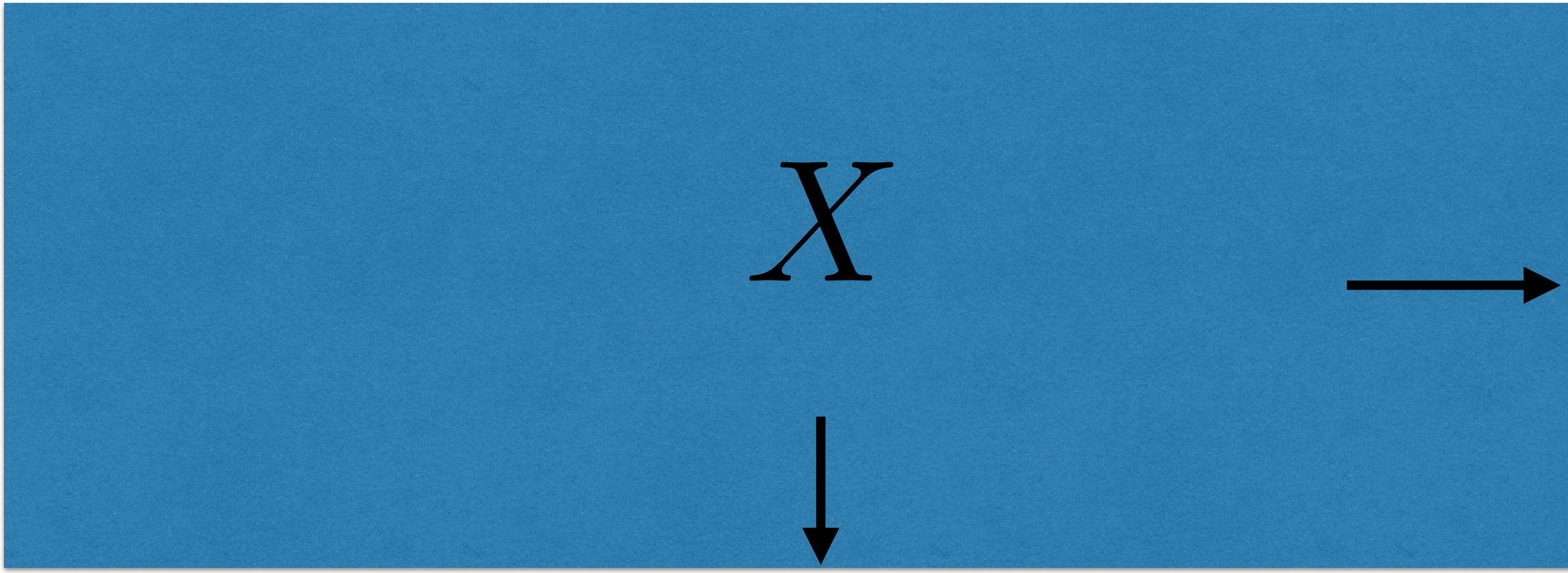
The Tall, **the Fat** and the Ugly



The Tall, **the Fat** and the Ugly



The Tall, the Fat and **the Ugly**



- d and n so large we can't even store in memory
- Only have time to be linear in $\text{size}(X) = n \times d$

I there any hope?

Pick a Random W

Pick a Random W

$$Y = X \times \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix}_K \Big/ \sqrt{d}$$

Why should Random Projections even work?!

Random Projection

- What does “it works” even mean?

Random Projection

- What does "it works" even mean?

  Distances between all pairs of data-points in low dim. projection is roughly the same as their distances in the high dim. space.

Random Projection

- What does "it works" even mean?

  Distances between all pairs of data-points in low dim. projection is roughly the same as their distances in the high dim. space.

That is, when K is "large enough", with "high probability", for all pairs of data points $i, j \in \{1, \ldots, n\}$,

$$(1-\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2 \leq \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \leq (1+\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Lets start with a one dimensional projection (K = 1)

$$y_t = \mathbf{x}_t^\top \mathbf{u} \quad \text{where each } \mathbf{u}[i] = \text{ random} \pm 1$$

- What is the expected value of:

1. $y_t - y_s$?
2. $(y_t - y_s)^2$?

Why should Random Projections even work?!

$$y_t - y_s = \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u}$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u}
\end{aligned}
$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u} \\
&= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbf{u}[k]
\end{aligned}
$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u} \\
&= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbf{u}[k]
\end{aligned}
$$

$$
\mathbb{E}\left[y_t - y_s\right] = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbb{E}\left[\mathbf{u}[k]\right] = 0
$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j]) \mathbf{u}[k] \cdot \mathbf{u}[j]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j]) \mathbf{u}[k] \cdot \mathbf{u}[j]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j] \ \mathbf{.} \ [\mathbf{u}[k] \cdot \mathbf{u}[j]]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$\mathbb{E}(y_t - y_s)^2$$

$$= \mathbb{E}\left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]\right)^2$$

$$= \mathbb{E}\left[\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbb{E}\left[\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$\mathbb{E}(y_t - y_s)^2$$

$$= \mathbb{E}\left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]\right)^2$$

$$= \mathbb{E}\left[\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbb{E}\left[\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 = \|\mathbf{x}_t - \mathbf{x}_s\|_2^2$$

Why should Random Projections even work?!

Hence for any $s, t \in \{1, \ldots, n\}$,

$$\mathbb{E}\left[|\mathbf{y}_s - \mathbf{y}_t|^2\right] = \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

Lets try ...

Why should Random Projections even work?!

Hence for any $s, t \in \{1, \ldots, n\}$,

$$\mathbb{E}\left[|\mathbf{y}_s - \mathbf{y}_t|^2\right] = \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

Lets try ...

Law of large numbers says that average over multiple draws is close to expectation

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

$$\mathbf{y}_t[k] = \mathbf{x}_t^\top \mathbf{u}_k / \sqrt{K} \quad \text{where each } \mathbf{u}_k[i] = \text{ random} \pm 1$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

$$\mathbf{y}_t[k] = \mathbf{x}_t^\top \mathbf{u}_k / \sqrt{K} \quad \text{where each } \mathbf{u}_k[i] = \text{ random} \pm 1$$

$$(\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \left(\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k\right)^2 / K$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

$$\mathbf{y}_t[k] = \mathbf{x}_t^\top \mathbf{u}_k / \sqrt{K} \quad \text{where each } \mathbf{u}_k[i] = \text{ random} \pm 1$$

$$(\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \left(\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k\right)^2 / K$$

$$\|\mathbf{y}_t - \mathbf{y}_s\|_2^2 = \sum_{k=1}^{K} (\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k\right)^2$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

$$\mathbf{y}_t[k] = \mathbf{x}_t^\top \mathbf{u}_k / \sqrt{K} \quad \text{where each } \mathbf{u}_k[i] = \text{ random} \pm 1$$

$$(\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \left(\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k\right)^2 / K$$

$$\|\mathbf{y}_t - \mathbf{y}_s\|_2^2 = \sum_{k=1}^{K} (\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k\right)^2$$

This is an average over K trials

Pick a Random W

Pick a Random W

$$Y = X \times \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix}_K \Big/ \sqrt{d}$$

Why should Random Projections even work?!

For any $\epsilon > 0$, if $K \approx \log(n/\delta)/\epsilon^2$, with probability $1 - \delta$ over draw of W, for all pairs of data points $i, j \in \{1, \ldots, n\}$,

$$(1-\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2 \leq \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \leq (1+\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2$$

Why should Random Projections even work?!

For any $\epsilon > 0$, if $K \approx \log(n/\delta)/\epsilon^2$, with probability $1 - \delta$ over draw of W, for all pairs of data points $i, j \in \{1, \ldots, n\}$,

$$(1-\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2 \leq \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \leq (1+\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2$$

Lets try ...

Why should Random Projections even work?!

For any $\epsilon > 0$, if $K \approx \log(n/\delta)/\epsilon^2$, with probability $1 - \delta$ over draw of W, for all pairs of data points $i, j \in \{1, \ldots, n\}$,

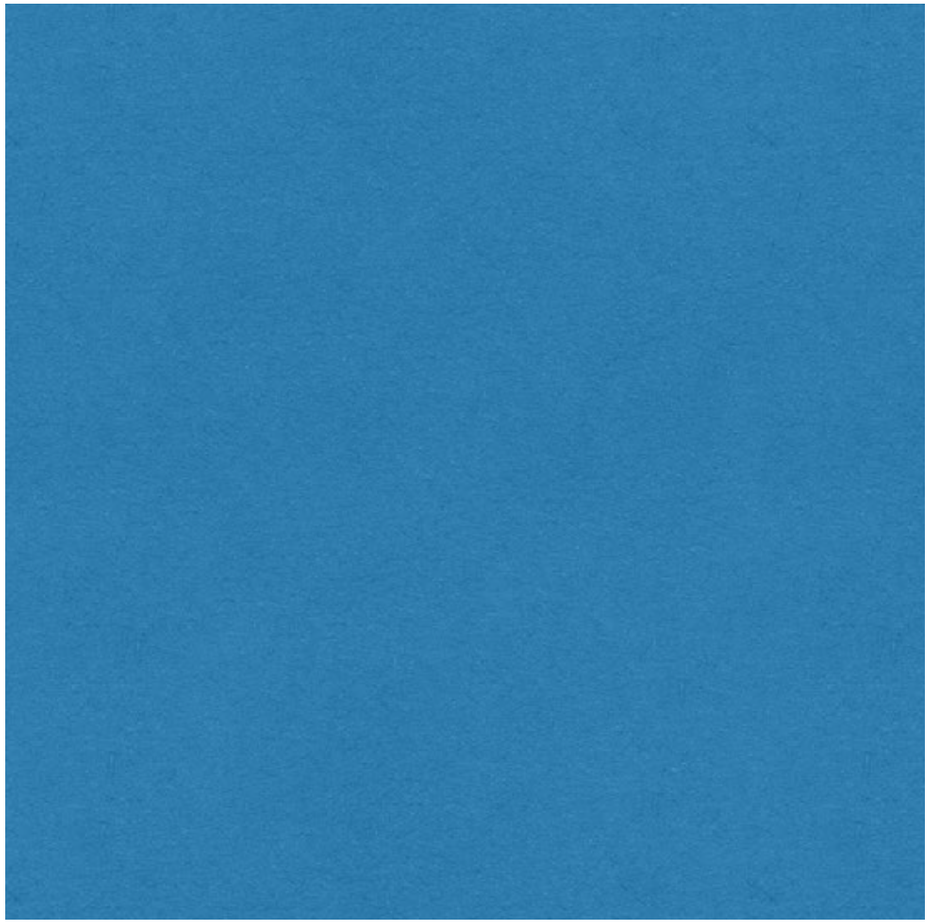
$$(1-\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2 \leq \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \leq (1+\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2$$

Lets try ...

This is called the Johnson-Lindenstrauss lemma or JL lemma for short.

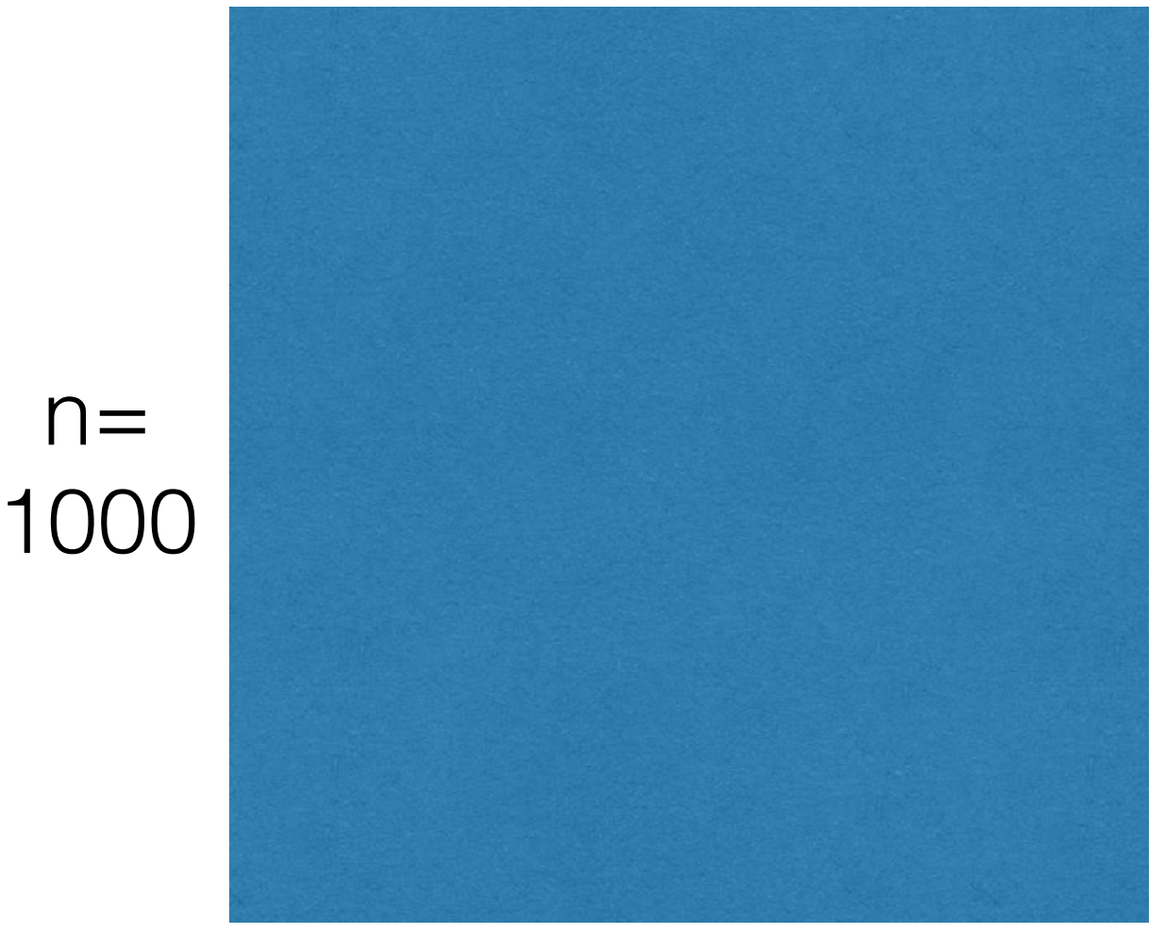
Why is this so Ridiculously Magical?

n=
1000



d = 1000

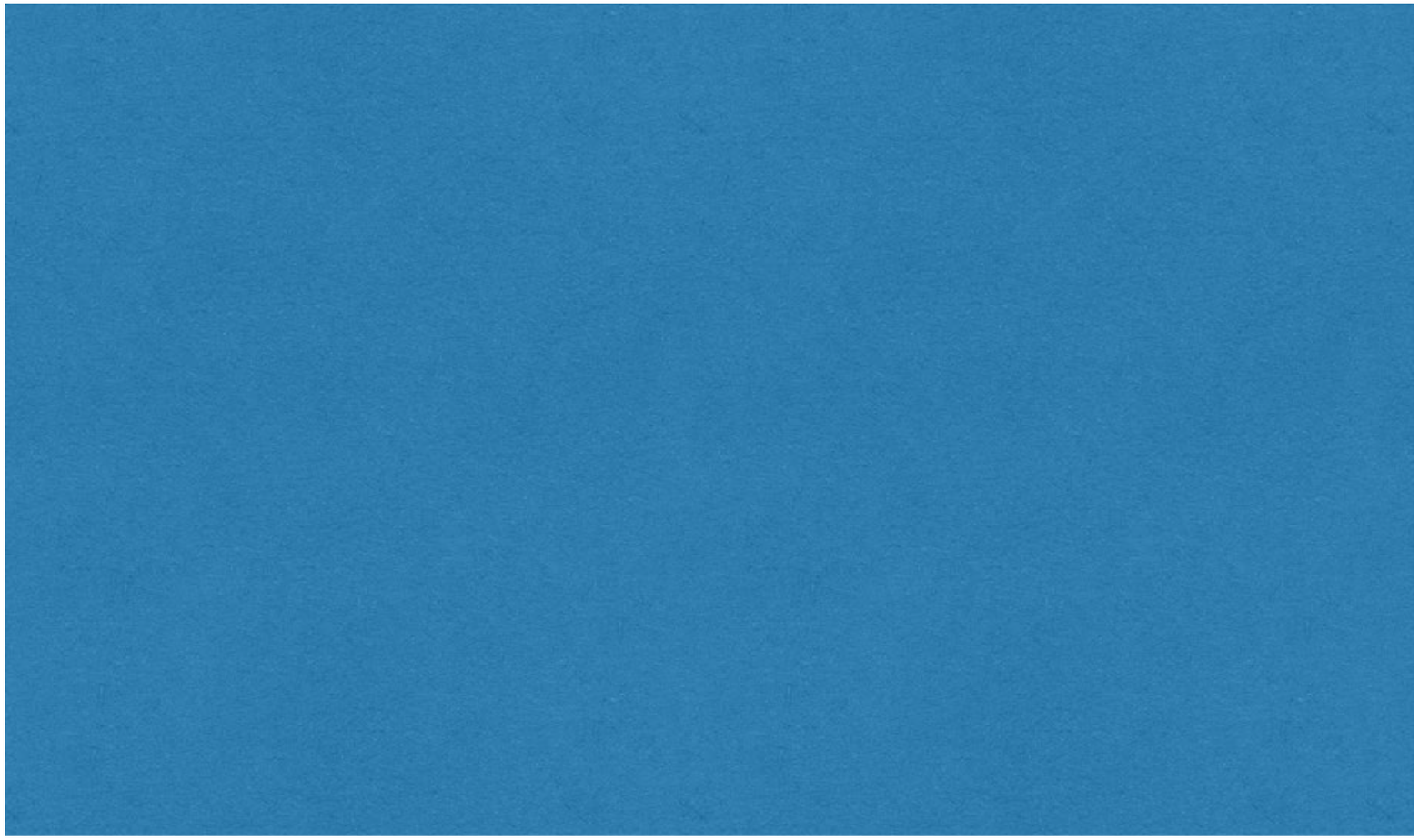
Why is this so Ridiculously Magical?



If we take $K = 69.1/\epsilon^2$, with probability 0.99 distances are preserved to accuracy ϵ

Why is this so Ridiculously Magical?

n=
1000

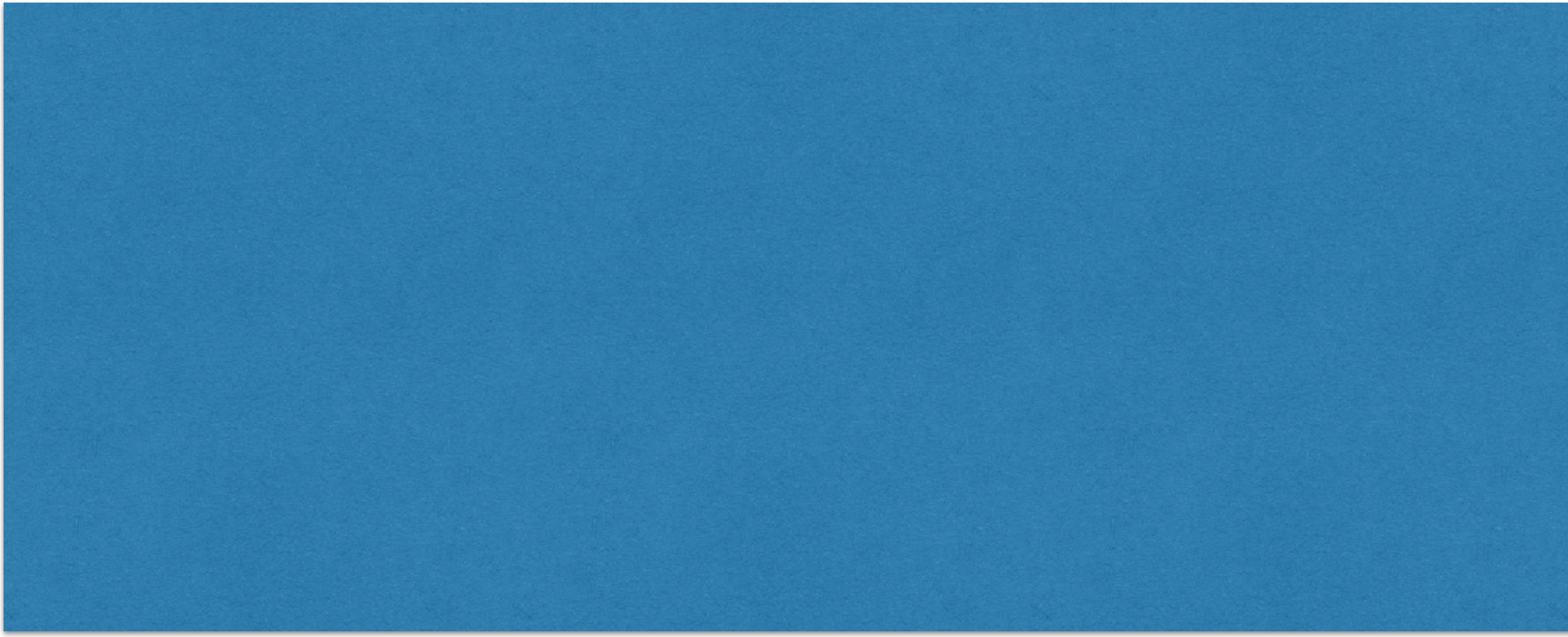


d = 10000

If we take $K = 69.1/\epsilon^2$, with probability 0.99 distances are preserved to accuracy ϵ

Why is this so Ridiculously Magical?

n=
1000



d = 1000000

If we take $K = 69.1/\epsilon^2$, with probability
0.99 distances are preserved to accuracy ϵ