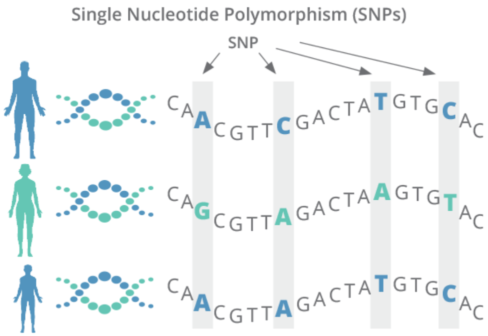
Unsupervised Dimensionality Reduction via PCA

David I. Inouye

Thursday, January 19, 2023

Very high-dimensional data is becoming ubiquitous

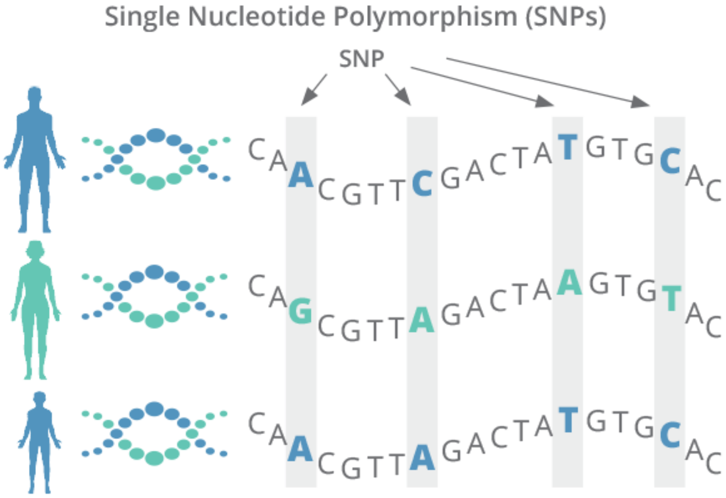
- Images (1 million pixels)
- Text (100k unique words)
- Genetics (4 million SNPs)
- Business data (12 million products)



<u>Why</u> dimensionality reduction? Lower computation costs

- Suppose original dimension is large like $\mathrm{d} = 100000$ (e.g., images, DNA sequencing, or text)

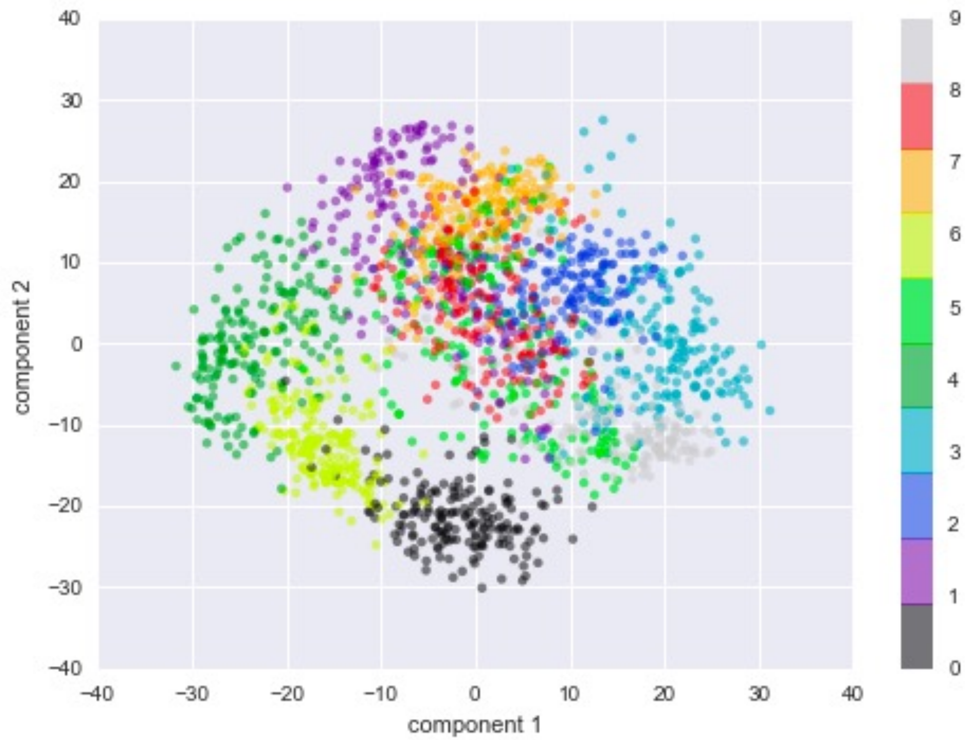
- If we reduce to $k = 100$ dimensions, the training algorithm can be sped up by $1000\times$



4-5 million SNPs in human genome.
https://www.diagnosticsolutionslab.com/tests/genomicinsight

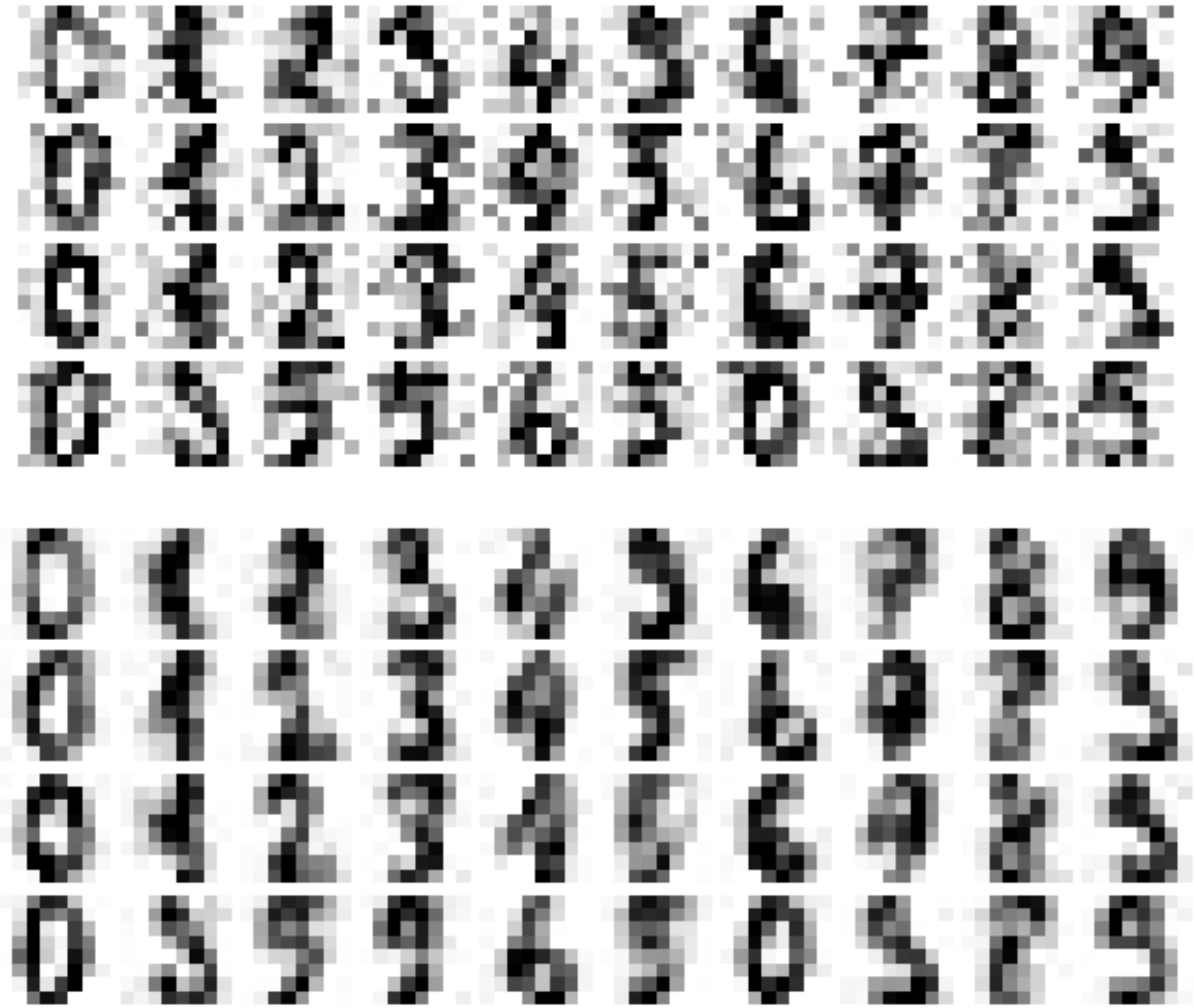
<u>Why</u> dimensionality reduction? Visualization

- Allows 2D scatterplot visualizations even of high-dimensional data (2D projection of digits)



https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

<u>Why</u> dimensionality reduction? Noise reduction via reconstruction



Outline of Principal Components Analysis (PCA)

1. Motivation for dimensionality reduction
2. Formal PCA problem: Min reconstruction
3. Derive PCA formulation for 1D
   - Least error 1D projection is orthogonal
   - Sum over all data points
4. Solution is based on truncated SVD
5. Alternative problem: Max variance

Review of linear algebra and introduction to numpy Python library

- See Jupyter notebook, which can be opened and run in Google Colab

Math: Principal Component Analysis (PCA) can be formalized as minimizing the linear reconstruction error of the data using only $k \le d$ dimensions

- PCA can be formalized as

$$\min_{\mathrm{Z,W}} \|X_c - ZW^T\|_F^2$$

  - where

$\mathrm{X_c} = \mathrm{X} - \mathbf{1}_n \mu_x^T \in \mathbb{R}^{n \times d}$ (centered input data)
$Z \in \mathbb{R}^{n \times k}$ (latent representation or "scores")
$W^T \in \mathbb{R}^{k \times d}$ (principal components)
$w_s^T w_t = 0, w_s^T w_s = \|w_s\|_2^2 = 1, \forall s, t$
(orthogonal constraint)

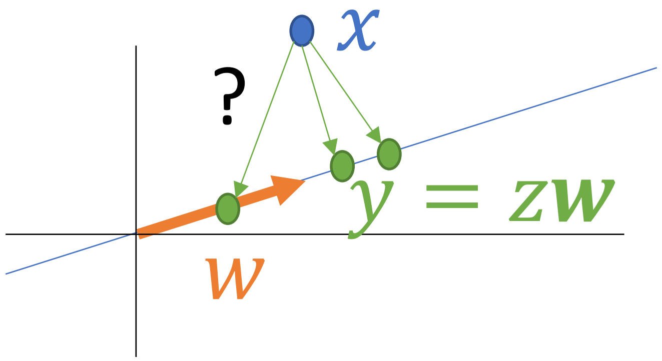
Math: Principal Component Analysis (PCA) can be formalized as minimizing the linear reconstruction error of the data using only $k \leq d$ dimensions

$$\min_{Z \in \mathbb{R}^{n \times k}, W \in \mathbb{R}^{d \times k}} \|X_c - ZW^T\|_F^2 \;\; \text{s.t. } W^T W = I_k$$

- Let's stare at this equation some more ☺
- Why is this dimensionality reduction?
- What does the orthogonal constraint mean?
- Why minimize the squared Frobenius norm?
- $\|X_c - ZW^T\|_F^2 = \sum_{i=1}^n \left\|\boldsymbol{x}_i^T - \boldsymbol{z}_i^T W^T\right\|_2^2 = \sum_{i=1}^n \|\boldsymbol{x_i} - W\boldsymbol{z}_i\|_2^2$
- For analysis, let's simplify to a single dimension (i.e., $k = 1$)
  - $\sum_{i=1}^n \|\boldsymbol{x_i} - z_i \boldsymbol{w}\|_2^2$ where z_i is a scalar

What is the best projection given a fixed subspace (line in 1D case)?

- If we are given $\boldsymbol{w}$, what is the best z (i.e. minimum reconstruction error) for a given $\boldsymbol{x}$?
  - $\min_z \|\boldsymbol{x} - z\boldsymbol{w}\|_2^2$



- The orthogonal projection!
  - $z = \boldsymbol{x}^T\boldsymbol{w} = \|\boldsymbol{x}\|\|\boldsymbol{w}\|\cos\theta = \|\boldsymbol{x}\|\cos\theta$
  - $z = \|\boldsymbol{x}\|\cos\theta = \mathrm{hyp}\cdot\frac{\mathrm{adj}}{\mathrm{hyp}} = \mathrm{adj}$
  - $z\boldsymbol{w}$ is a scaled vector along the line defined by $\boldsymbol{w}$

Thus, we can simplify to only minimizing over W

$$\min_{\boldsymbol{z},\boldsymbol{w}:\|\boldsymbol{w}\|_2=1} \sum_{i=1}^{n} \|\boldsymbol{x_i} - z_i \boldsymbol{w}\|_2^2 = \min_{\boldsymbol{w}:\|\boldsymbol{w}\|_2=1} \sum_{i=1}^{n} \left\|\boldsymbol{x_i} - (\boldsymbol{x}_i^T \boldsymbol{w})\boldsymbol{w}\right\|_2^2$$

- Now we can return to the Frobenius norm:

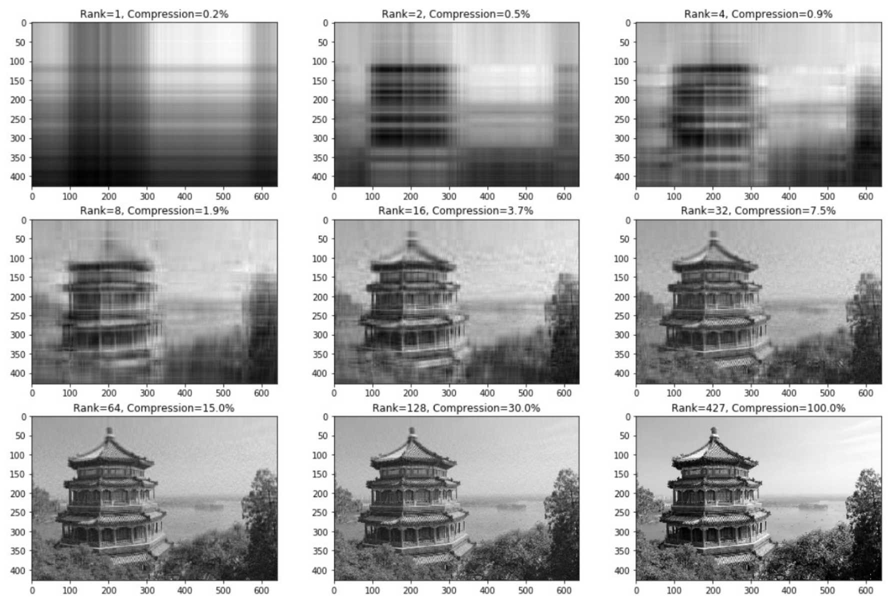
$$\min_{\boldsymbol{w}:\|\boldsymbol{w}\|_2=1} \left\|X_c - \boldsymbol{z}\boldsymbol{w}^{\boldsymbol{T}}\right\|_F^2 \quad \text{where} \quad \boldsymbol{z} = X_c \boldsymbol{w}$$

- What is $\boldsymbol{z}\boldsymbol{w}^{\boldsymbol{T}}$? Have we seen something like this before?
- This is the best low-rank approximation to X_c, which is given by the SVD!
  - $\boldsymbol{w} = \boldsymbol{v}_1$ and $\boldsymbol{z} = \sigma_1 \boldsymbol{u}_1$, where $\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1$ are the first singular value, left singular vector and right singular vector respectively.

For $k \geq 1$, the PCA solution is the top k right singular vectors

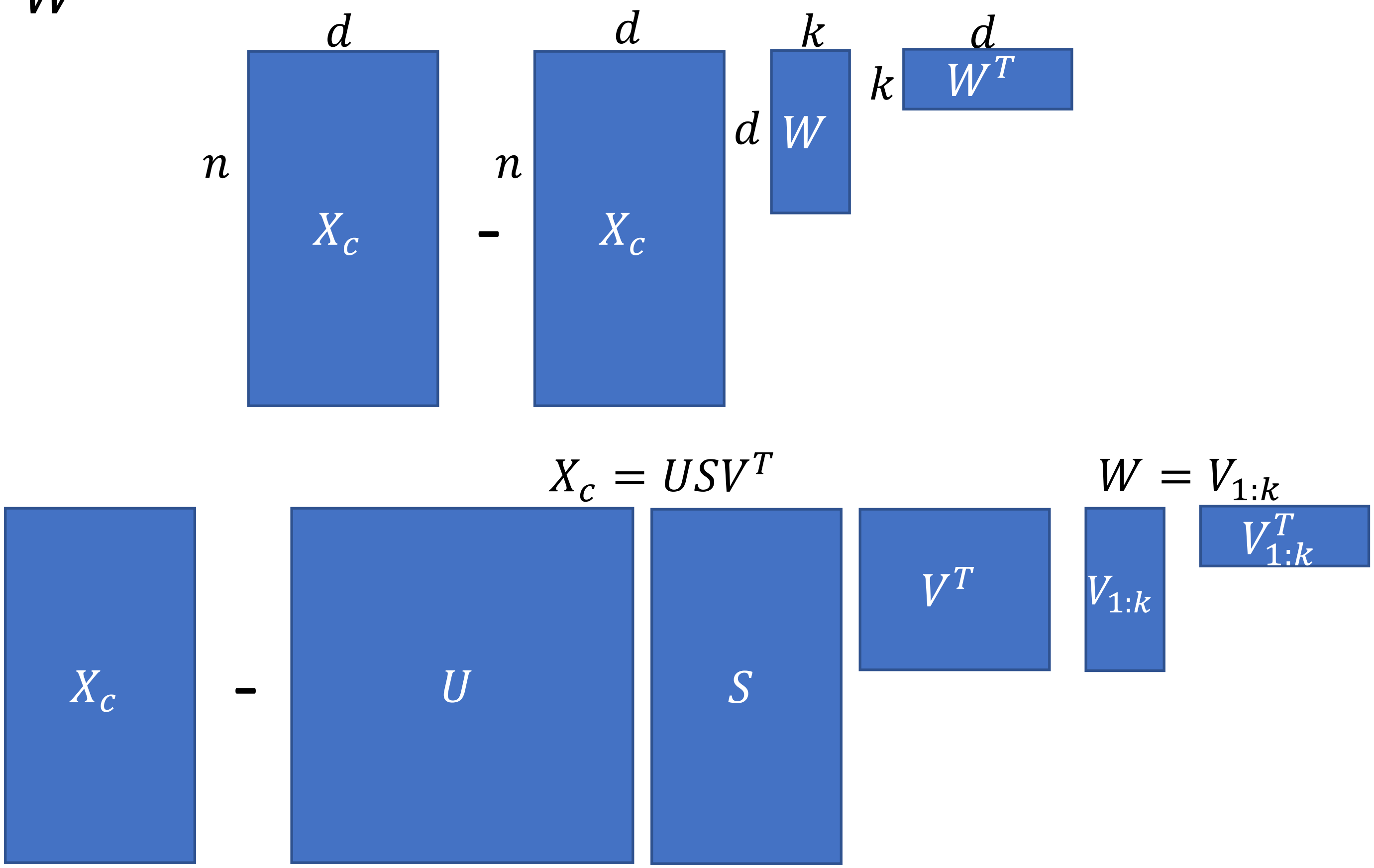
- If $X_c = USV^T$, then the general solution is
$$W^* = V_{1:k}$$

- Remember: SVD is best k dim. approximation



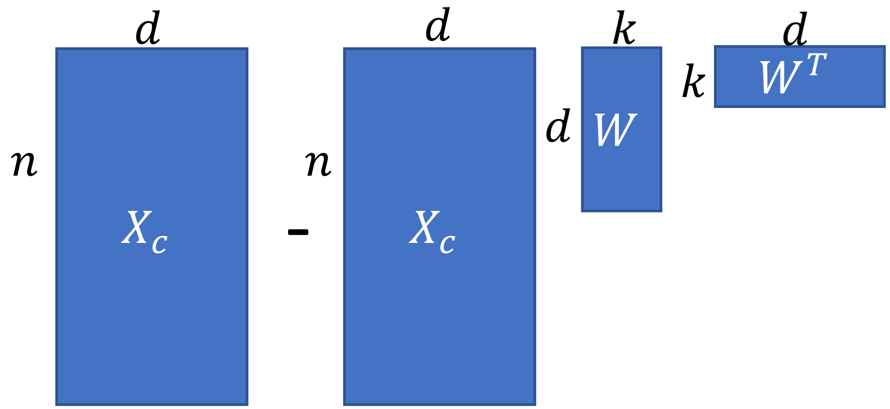
Check: The solution reveals the truncated SVD as best approximation

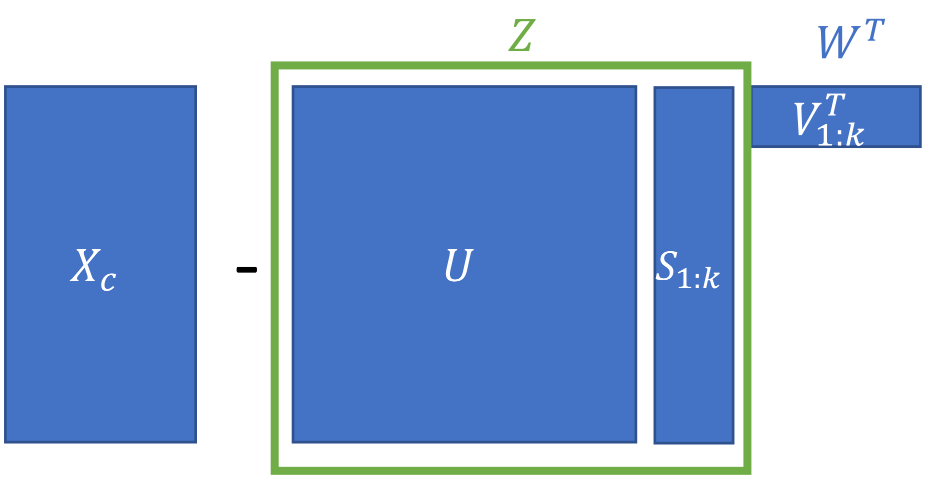
$$\min_{W} \|X_c - (X_c W) W^T\|_F^2 \quad \text{s.t.} \quad W^T W = I$$



Check: The solution reveals the truncated SVD as best approximation

$$\min_W \|X_c - (X_c W)W^T\|_F^2 \;\; \text{s.t.} \;\; W^T W = I$$



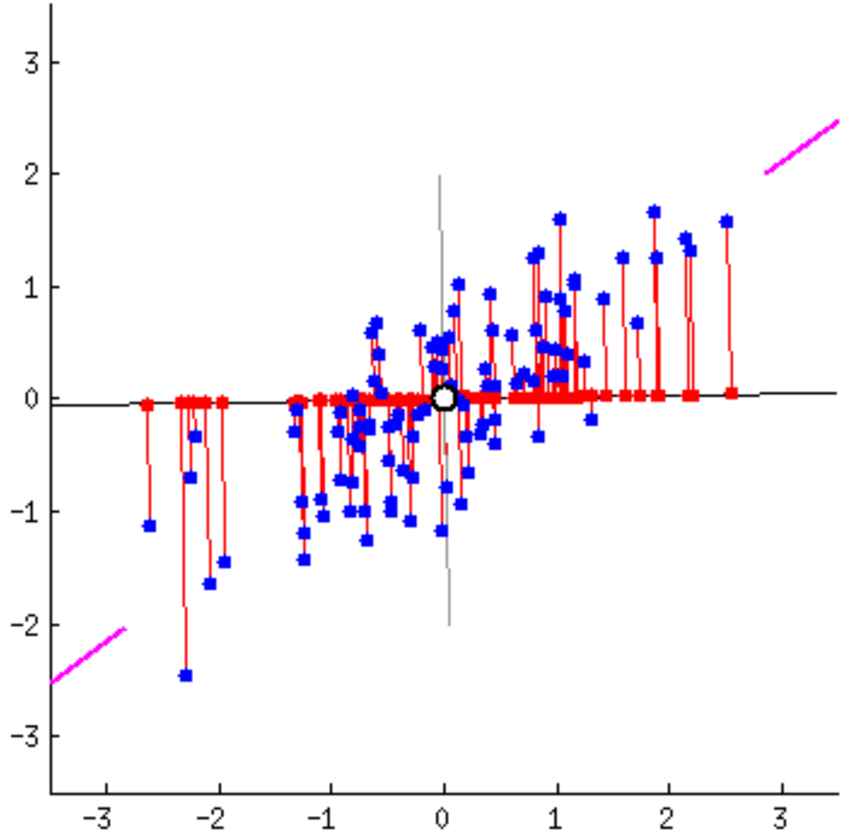


Top k truncated SVD

Intuition: Principal component analysis finds the **best linear projection** onto a lower-dimensional space

$$\min_{\boldsymbol{w}:\|\boldsymbol{w}\|_2=1} \left\| X_c - \boldsymbol{z}\boldsymbol{w}^{\boldsymbol{T}} \right\|_F^2$$

where $\boldsymbol{z} = X_c \boldsymbol{w}$



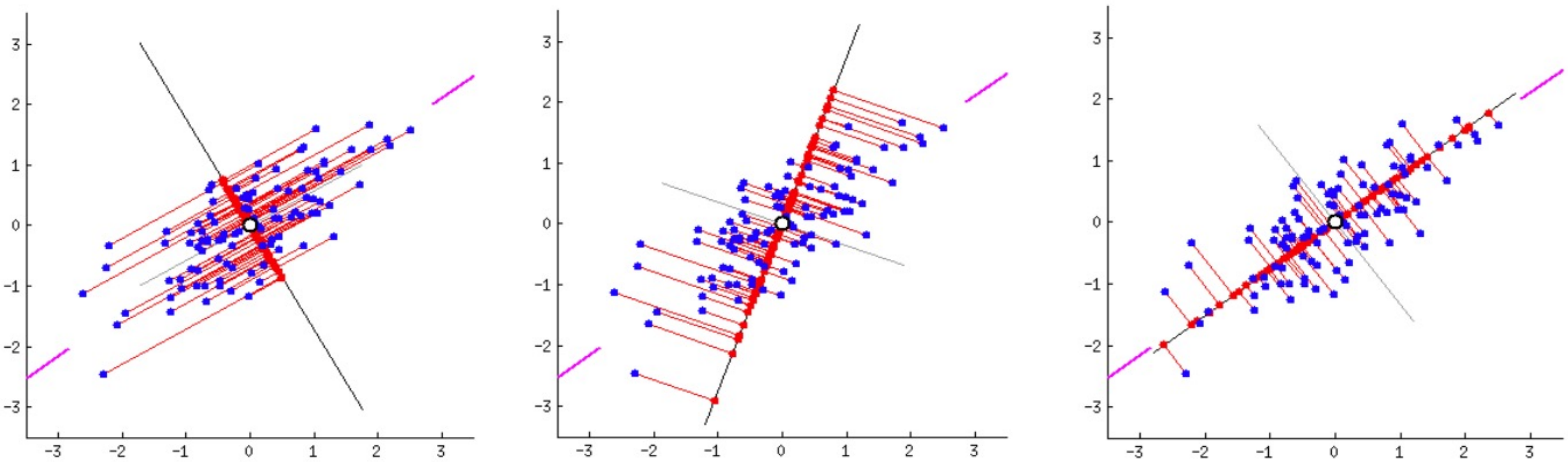
2D to 1D projection: Red lines show the projection error onto 1D lines. PCA finds the line that has the smallest projection error (in this example, when it aligns with the purple).

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

Minimizing reconstruction error (red lines) is equivalent to maximizing the variance of projection (spread of red points)

Max reconstruction error
Min variance

Min reconstruction error
Max variance



Equivalent solutions: The solution to both problems is the top k right singular vectors of X_c

- Minimize reconstruction error
$$\min_{W:W^TW=I_k} \|X_c - (X_cW)W^T\|_F^2$$
  - Singular value decomposition (SVD) of $X_c = USV^T$
  - Solution: $W^* = V_{1:k}$

- Maximize variance of latent projection (equivalent solution)
$$\max_{W:W^TW=I_k} \mathrm{Tr}\left(W^T\hat{\Sigma}W\right)$$
  - where $\hat{\Sigma} \coloneqq \frac{1}{n}X_c^TX_c$ is the covariance matrix
  - $n\hat{\Sigma} = X_c^TX_c = (USV^T)^T(USV^T) = (VSU^T)(USV^T) = VS(U^TU)SV^T = VS^2V^T = Q\Lambda Q^T$
  - Solution: $W^* = Q_{1:k} \equiv V_{1:k}$!

Recap: Principal Components Analysis (PCA)

1. Motivation for dimensionality reduction
2. Formal PCA problem: Min reconstruction
3. Derive PCA formulation for 1D
   - Least error 1D projection is orthogonal
   - Sum over all data points
4. Solution is based on truncated SVD
5. Alternative viewpoint: Max variance
   - Derive equivalence
   - Derive equivalent solutions

Demo of PCA via sklearn (time permitting)

- Random projections vs PCA projections
- Visualizations of
  - Minimum reconstruction error
  - Maximum variance
  - Explained variance based on k
- Code examples
  - Digits
  - Eigenfaces

Questions?

Optional extra derivation slides

How is PCA similar or different than the following maximization problem?

- Minimize reconstruction error
$$\min_{W:W^TW=I_k} \|X_c - (X_cW)W^T\|_F^2$$
- Alternative problem
$$\max_{W:W^TW=I_k} \operatorname{Tr}(W^TX_c^TX_cW)$$
- $\operatorname{Tr}(W^TX_c^TX_cW) = \operatorname{Tr}\big((X_cW)^T(X_cW)\big)$
- $= \operatorname{Tr}(Z^TZ)$
- $= \sum_{j=1}^k \boldsymbol{z}_j^T\boldsymbol{z}_j$
- $= n\sum_{j=1}^k \frac{1}{n}\sum_{i=1}^n z_{i,j}^2$
- $= n\sum_{j=1}^k \sigma_{z,j}^2$ where $\sigma_{z,j}^2$ is the variance of the j-th latent dimension
- Given this, what does the optimization problem **mean**?
- Answer: This objective maximizes the sum of variances of the data projected onto W.

1D derivation of min error equivalent to max variance

- First step: Simplify squared distance
- $\|x_i - (x_i^T w)w\|_2^2$
- $= (x_i - (x_i^T w)w)^T (x_i - (x_i^T w)w)$
- $= x_i^T x_i - 2(x_i^T w)w^T x_i + (x_i^T w)^2 w^T w$
- $= \|x_i\|^2 - 2(x_i^T w)^2 + (x_i^T w)^2 \|w\|^2$
- $= \|x_i\|^2 - (x_i^T w)^2$

1D derivation of min error equivalent to max variance

- Equivalence of optimization in 1D
- $\arg\min_w \sum_i \left\| x_i - (x_i^T w) w \right\|_2^2$
- $= \arg\min_w \sum_i \|x_i\|^2 - \left(x_i^T w\right)^2 = \arg\min_w \sum_i -\left(x_i^T w\right)^2$
- $= \arg\max_w \frac{1}{n} \sum_i \left(x_i^T w\right)^2 = \arg\max_w \frac{1}{n} \sum_i z_i^2$
- $= \arg\max_w \ \sigma_z^2$

Note z is already centered so mean of squares is variance

Therefore, we can reformulate the problem as maximizing the variance

- Let's rewrite this last term
  - $\sigma_z^2 = \frac{1}{n}\sum_i (z_i)^2 = \frac{1}{n}\sum_i \left(x_i^T w\right)^2 = \frac{1}{n}(X_c w)^T (X_c w) = w^T \left(\frac{1}{n} X_c^T X_c\right) w = w^T \hat{\Sigma} w$
- Thus, our problem can be formulated as:
  - $\max_{w:\|w\|=1} w^T \hat{\Sigma} w$
- The solution is the eigenvector q_1 of $\hat{\Sigma} = Q\Lambda \mathrm{Q}^{\mathrm{T}}$ corresponding to the largest eigenvalue λ_1

$$w^* = q_1$$

For $k > 1$, we maximize the sum of variances for each latent dimension

- More generally we can formulate this as:
  - $\max_{W:W^TW=I_k} \sum_{j=1}^{k} \sigma_{z_j}^2$
  - $= \max_{W:W^TW=I_k} \sum_{j=1}^{k} w_j^T \hat{\Sigma} w_j$
  - $= \max_{W:W^TW=I_k} \mathrm{Tr}\left(W^T \hat{\Sigma} W\right)$
  - $= \max_{W:W^TW=I_k} \frac{1}{n} \mathrm{Tr}(W^T X_c^T X_c W)$

- The solution is the top k eigenvectors of $\hat{\Sigma} = Q\Lambda Q^T$
  - $W^* = Q_{1:k}$