Dimensionality Reduction

- We have some data $X \in \mathbb{R}^{N \times D}$
- $D$ may be huge, etc.
- We would like to find a new representation $Z \in \mathbb{R}^{N \times K}$ where $K << D$.
  - For computational reasons.
  - To better understand (e.g., visualize) the data.
  - For compression.
  - ...
- We will restrict ourselves to linear transformations for the time being.
Example

- In this dataset, there are only 3 degrees of freedom: horizontal and vertical translations, and rotations.
- Yet each image contains 784 pixels, so $X$ will be 784 elements wide.
Abstract Visualization
What is a Good Transformation?

- Goal is to find good directions $u$ that preserves “important” aspects of the data.
- In a linear setting: $z = x^T u$
- This will turn out to be the top-K eigenvalues of the data covariance.
- Two ways to view this:
  1. Find directions of \textit{maximum variation}
  2. Find projections that \textit{minimize reconstruction error}
Principal Component Analysis (Maximum Variance)

\[
\text{maximize } \frac{1}{2N} \sum_{n=1}^{N} (u_1^T x_n - u_1^T \bar{x}_n)^2
\]

\[= u_1^T S u_1\]

i.e., variance of the projected data

where the sample mean and covariance are given by:

\[
\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n
\]

\[
S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T
\]
Finding $u_1$

- We want to maximize $u_1^T Su_1$

subject to $||u_1|| = 1$
(since we are finding a direction)

- Use Lagrange multiplier $\alpha_1$ to express this as

$$u_1^T Su_1 + \alpha_1 (1 - u_1^T u_1)$$
Finding $u_1$

- Take derivative and set to 0

$$Su_1 - \alpha_1 u_1 = 0$$
$$Su_1 = \alpha_1 u_1$$

- So $u_1$ is an eigenvector of $S$ with eigenvalue $\alpha_1$

- In fact it must be the eigenvector with maximum eigenvalue, since this minimizes the objective.
Finding $u_2$

maximize $u_2^T S u_2$

subject to $||u_2|| = 1$

$u_2^T u_1 = 0$

Lagrange form: $u_2^T S u_2 + \alpha_2 (1 - u_2^T u_2) - \beta u_2^T u_1$

Finding $\beta$:

$$\frac{\partial}{\partial u_2} = S u_2 - \alpha_2 u_2 - \beta u_1 = 0$$

$$\implies u_1^T S u_2 - \alpha_2 u_1^T u_2 - \beta u_1^T u_1 = 0$$

$$\implies \alpha_1 u_1^T u_2 - \alpha_2 u_1^T u_2 - \beta u_1^T u_1 = 0$$

$$\implies \alpha_1 \cdot 0 - \alpha_2 \cdot 0 - \beta \cdot 1 = 0$$

$$\implies \beta = 0$$
Finding $u_2$

maximize $u_2^T S u_2$

subject to $||u_2|| = 1$

$u_2^T u_1 = 0$

Lagrange form: $u_2^T S u_2 + \alpha_2 (1 - u_2^T u_2) - \beta u_2^T u_1$

Finding $\alpha_2$:

$$\frac{\partial}{\partial u_2} = S u_2 - \alpha_2 u_2 = 0$$

$\implies S u_2 = \alpha_2 u_2$

So $\alpha_2$ must be the second largest eigenvalue of $S$. 
PCA in General

- We can compute the entire PCA solution by just computing the eigenvectors with the top-k eigenvalues.
- These can be found using the singular value decomposition of $S$. 
How do we choose the number of components?

- Look at the spectrum of covariance, pick $K$ to capture most of the variation.
- More principled: Bayesian treatment (beyond this course).
Demo

- Eigenfaces
Principal Component Analysis (Minimum Reconstruction Error)

- We can also think of PCA as minimizing the reconstruction error of the compressed data.

\[
\text{minimize } = \frac{1}{2N} \sum_{n=1}^{N} ||x_n - \hat{x}_n||^2
\]

- We will omit the details for now, but the key is that we define some K-dimensional basis such that:

\[
\hat{x} = Wx + \text{const}
\]

- The solution will turn out to be the same as the minimum variance formulation.
Reconstruction

- PCA learns to represent vectors in terms of sums of basis vectors.
- For images, e.g.,

\[
\text{Image} = a_1 + a_2 + a_3 + \ldots + a_{100} + \ldots
\]
PCA for Compression

D in this slide is the same as K in the previous slides

321x481 image, D is the number of basis vectors used
Summary
Thank You ;-)