Tutorial 4: PCA

Principal Component Analysis

TA: Paul-Edouard Sarlin psarlin@ethz.ch
Lossy vs Lossless compression

- Data $x \in A$ e.g. an image
- Compressor $f: A \rightarrow B$
- $f(x) = \text{compressed version such that } \text{size}(f(x)) < \text{size}(x)$
- $f^{-1}(f(x)) = \text{decoding of the compressed } x$
- Reconstruction error $E = \|x - f^{-1}(f(x))\|$

<table>
<thead>
<tr>
<th>Lossless</th>
<th>Lossy</th>
</tr>
</thead>
<tbody>
<tr>
<td>if $E = 0$</td>
<td>if $E \neq 0$</td>
</tr>
<tr>
<td></td>
<td>But good if $E$ small enough</td>
</tr>
</tbody>
</table>

PCA
Principal Component Analysis - PCA

• **Non-parametric** method of extracting relevant information from data

• **Orthogonal linear projection** of high dimensional data onto low dimensional subspace

\[ f(x) = Ux \]

Properties:

1. **Reconstruct the data well**: minimize error \( E \)

2. **Maximize information**: maximize the total variance of the encoding \( f(x) \)
How to calculate $U$?

• We have a collection of $N$ data samples $x_i \in \mathbb{R}^D$
• We can fit a normal distribution $\mathcal{N} (\mu, \Sigma)$ to the data:
  \[
  \mu = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \Sigma = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T
  \]
• $U = \textit{eigenmatrix}$ of $\Sigma$, such that $\Sigma = U\Lambda U^T$ (orthogonal!)
• For PCA: $U_K$ = first $K$ eigenvectors of $\Sigma = [u_1 \ldots u_K]$
• Then $\text{PCA}(x_i) = f(x_i) = U_K^T (x_i - \mu)$
• $K$ principal components = directions with largest variance
• Large compression if $K \ll D$
Using SVD vs Eigendecomposition

- Let $X$ be the $D \times N$ data matrix: $X = [x_1 \ldots x_N]$
- Let $\bar{X}$ be the centered data matrix $\bar{X} = X - \mu$
- Apply SVD: $\bar{X} = USV^T$ where $U^T U = I_D$ and $V^T V = I_N$
- Thus $\Sigma = \bar{X} \bar{X}^T = USV^T VSU^T = US^2 U^T$
- Thus we can compute PCA with either SVD or Eigendecomposition
PCA on faces

• Now $x_i$ = images of human faces
• $x_i \in \mathbb{R}^D$ with $D$ = number of pixels = width x height

• AT&T face database: 40 people, 10 expressions each
Compute:

• Mean $\mu \in \mathbb{R}^D$

• Covariance $\Sigma \in \mathbb{R}^{D \times D}$
PCA on faces

• First 10 eigenvectors $u_{1:K}$ ordered by decreasing eigenvalues
PCA on faces

- First 10 eigenvectors $u_{1:K}$ ordered by decreasing eigenvalues

- Take $K = 2$

original face

$\text{reconstructed} = \text{mean face} + c_1 + c_2$

mean face
eigen face 1
eigen face 2
PCA compression

- We have $D = 68 \times 56 = 3808$
- If $K = 100$ then each face is represented by only 100 values
- That’s a 38x compression!

![Original vs Reconstructed Faces](image-url)
PCA compression

\[ K = 50 \]

\[ K = 200 \]
Application to Face Detection

George (s38)

• Can you compute the x coordinate of George’s head?
Application to Face Detection

• Steps:
  – Compute eigenfaces (use only the first 20 people)
  – Compress each patch of the image using a sliding window
  – Evaluate the compression error using SSD
  – The patch with the lowest error is George!
Application to Face Detection

George (s38)