Recap

• (Ch 10) Data in high dimensions
  • Visualizing data
  • Summarizing data

Today

• (Ch 10) Data in high dimensions
  • Dimensionality reduction
  • Principal components analysis
Covariance matrix of multidimensional data

- Given a dataset \( \{\mathbf{x}\} \) of \( N \) \( d \)-dimensional vectors \( \mathbf{x}_i \), the covariance matrix is a \( d \times d \) matrix

\[
\text{Covmat}(\{\mathbf{x}\}) = \frac{\sum_i (\mathbf{x}_i - \text{mean}(\{\mathbf{x}\}))(\mathbf{x}_i - \text{mean}(\{\mathbf{x}\}))^T}{N}
\]

- Properties
  - The \((j, k)\) entry of \( \text{Covmat}(\{\mathbf{x}\}) \) is \( \text{cov}(\{\mathbf{x}^{(j)}\}, \{\mathbf{x}^{(k)}\}) \)
  - The \((j, j)\) entry of \( \text{Covmat}(\{\mathbf{x}\}) \) is \( \text{var}(\{\mathbf{x}^{(j)}\}) \)
  - \( \text{Covmat}(\{\mathbf{x}\}) \) is symmetric
Translation properties

• Translating the data translates the mean

\[ \text{mean}(\{x + c\}) = \text{mean}(\{x\}) + c \]

• Translating the data leaves the covariance matrix unchanged

\[ \text{Covmat}(\{x + c\}) = \text{Covmat}(\{x\}) \]
Linear transformation properties

- Linearly transforming the data linearly transforms the mean
  \[ \text{mean}(\{Ax\}) = A \text{mean}(\{x\}) \]

- Linearly transforming the data changes the covariance matrix
  \[ \text{Covmat}(\{Ax\}) = A \text{Covmat}(\{x\}) A^T \]
Dimensionality reduction: 2D to 1D example
Step 1: subtract mean
Step 2: apply linear transformation

[Diagram showing data points before and after rotation to diagonalize covariance]
Step 3: drop component(s)
Principal components analysis (PCA)

We will reduce the dimensionality of dataset \{x\} from \(d\) to \(s\)

- Step 1: define \{m\} such that \(m_i = x_i - \text{mean}({x})\)

- Step 2: define \{r\} such that \(r_i = U^T m_i\)
  where \(\Lambda = U^T \text{Covmat}({x}) U\) is the diagonalization of \(\text{Covmat}({x})\) with the eigenvalues sorted in decreasing order

- Step 3: define \{p\} such that each \(p_i\) is \(r_i\) with the last \(d - s\) components zeroed out (or discarded)
What happens to the mean?

• Step 1:
  \[
  \text{mean(}\{\mathbf{m}\}\text{)} = \text{mean(}\{\mathbf{x}\}\text{)} - \text{mean(}\{\mathbf{x}\}\text{)} = 0
  \]

• Step 2:
  \[
  \text{mean(}\{\mathbf{r}\}\text{)} = U^T \text{mean(}\{\mathbf{m}\}\text{)} = U^T 0 = 0
  \]

• Step 3:
  \[
  \text{mean(}\{\mathbf{p}\}\text{)} = \text{mean(}\{\mathbf{r}\}\text{)} = 0
  \]
What happens to the covariance matrix?

- Step 1:
  \[
  \text{Covmat}([\mathbf{m}]) = \text{Covmat}([\mathbf{x}])
  \]

- Step 2:
  \[
  \text{Covmat}([\mathbf{r}]) = U^T \text{Covmat}([\mathbf{m}]) U = \Lambda
  \]

- Step 3:
  \[
  \text{Covmat}([\mathbf{p}]) \text{ is } \Lambda \text{ with the last } d - s \text{ diagonal terms zeroed out}
  \]
Mean square error of the projection (step 3)

\[
\frac{1}{N} \sum_i \| r_i - p_i \|^2 = \frac{1}{N} \sum_i \sum_{j=s+1}^{d} (r_i^{(j)})^2 = \sum_{j=s+1}^{d} \frac{1}{N} \sum_i (r_i^{(j)})^2 \\
= \sum_{j=s+1}^{d} \text{var}(\{r^{(j)}\}) \quad \text{since mean}(\{r^{(j)}\}) = 0 \\
= \sum_{j=s+1}^{d} \lambda_j \quad \text{since the variances are diagonal entries of } \text{Covmat}(S-1) = \Lambda
\]

The mean square error is the sum of the smallest \(d - s\) eigenvalues in \(\Lambda\)
PCA: iris dataset example

- The Iris dataset is a famous dataset consisting of measurements of three different varieties of iris flowers
  - Iris-setosa
  - Iris-versicolor
  - Iris-virginica

- There are 4 measurements per item
  - Sepal length (cm)
  - Sepal width (cm)
  - Petal length (cm)
  - Petal width (cm)

- See today’s Jupyter notebook
Reconstructing the data

- Given the projected dataset \( \{p\} \) and \( \text{mean}(\{x\}) \), we can approximately reconstruct the original dataset as \( \{\hat{x}\} \)

\[ \hat{x}_i = Up_i + \text{mean}(\{x\}) \]

- Notice that each reconstructed data item \( \hat{x}_i \) is \( \text{mean}(\{x\}) \) plus a linear combination of the columns of \( U \) weighted by the entries in \( p_i \)

- The columns of \( U \) are the normalized eigenvectors of \( \text{Covmat}(\{x\}) \) and are called the principal components of the data \( \{x\} \)
End-to-end mean square error

- Each $x_i$ becomes $r_i$ by translation and rotation
- Each $p_i$ becomes $\hat{x}_i$ by the opposite rotation and translation
- Therefore, the end-to-end mean square error is

$$\frac{1}{N} \sum_i \|\hat{x}_i - x_i\|^2 = \frac{1}{N} \sum_i \|r_i - p_i\|^2 = \sum_{j=s+1}^{d} \lambda_j$$

where $\lambda_{s+1}, ..., \lambda_{d}$ are the smallest $d - s$ eigenvalues of $\text{Covmat}(\{x\})$
PCA: Japanese face dataset example

- The dataset consists of 213 images of Japanese women
- Each image is grayscale and has $64 \times 64$ resolution
- We can treat each image as a vector of dimension $d = 4096$
How quickly do the eigenvalues drop off?
What do the principal components look like?

- The mean face is blurry
- The first few principal components capture
  - Shape of hair
  - Height of face
  - Height of eyebrows
  - Etc.
What do the reconstructions look like?

<table>
<thead>
<tr>
<th>Number of principal components</th>
<th>mean</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reconstructions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>