Data-driven Methods: Faces

Portrait of Piotr Gibas
The Power of Averaging
8-hour exposure
Sir Francis Galton 1822-1911

Multiple Individuals

Average Images in Art

“60 passagers de 2e classe du metro, entre 9h et 11h” (1985)
Krzysztof Pruszkowski

“Spherical type gasholders” (2004)
Idris Khan
“100 Special Moments” by Jason Salavon

Why blurry?
Object-Centric Averages by Torralba (2001)

Manual Annotation and Alignment

Average Image

Slide by Jun-Yan Zhu
Computing Means

Two Requirements:
• Alignment of objects
• Objects must span a subspace

Useful concepts:
• Subpopulation means
• Deviations from the mean
Images as Vectors

\[ n \times m = n^*m \]
Vector Mean: Importance of Alignment

Let $n \times m$ images be represented as $n \times m$ matrices. The mean image can be calculated as follows:

$$\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} = \text{mean image}$$
How to align faces?

http://www2.imm.dtu.dk/~aam/datasets/datasets.html
Shape Vector

Provides alignment!
Appearance Vectors vs. Shape Vectors

Appearance Vector
- 200*150 pixels (RGB)
- Vector of 200*150*3 Dimensions
- Requires Annotation
- Provides alignment!

Shape Vector
- 43 coordinates (x,y)
- Vector of 43*2 Dimensions

Slide by Kevin Karsch
Average Face

1. Warp to mean shape
2. Average pixels

Objects must span a subspace

This is also a valid object
Example

Does not span a subspace
Subpopulation means

Examples:
- Male vs. female
- Happy vs. sad
- Average Kids
- Happy Males
- Etc.
- http://www.faceresearch.org

Average male

Average happy male

Average kid

Average female
Average Women of the world
The space of all face images

- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images
Principal Component Analysis

Suppose the data points are arranged as above

- Idea—fit a line, classifier measures distance to line

\[ \mathbf{x} \rightarrow ((\mathbf{x} - \mathbf{x}) \cdot \mathbf{v}_1, (\mathbf{x} - \mathbf{x}) \cdot \mathbf{v}_2) \]

What does the \( \mathbf{v}_2 \) coordinate measure?
- distance to line
- use it for classification—near 0 for orange pts

What does the \( \mathbf{v}_1 \) coordinate measure?
- position along line
- use it to specify which orange point it is

\( \mathbf{v}_1 \) is the mean of the orange points
Principal Component Analysis

How to find $v_1$ and $v_2$?

Dimensionality reduction

- We can represent the orange points with *only* their $v_1$ coordinates
  - since $v_2$ coordinates are all essentially 0
- This makes it much cheaper to store and compare points
- A bigger deal for higher dimensional problems
Linear subspaces

Consider the variation along direction $\mathbf{v}$ among all of the orange points:

$$\text{var}(\mathbf{v}) = \sum_{x} \| (x - \bar{x})^T \cdot \mathbf{v} \|^2$$

What unit vector $\mathbf{v}$ minimizes $\text{var}$?

$$\mathbf{v}_2 = \min_{\mathbf{v}} \{ \text{var}(\mathbf{v}) \}$$

What unit vector $\mathbf{v}$ maximizes $\text{var}$?

$$\mathbf{v}_1 = \max_{\mathbf{v}} \{ \text{var}(\mathbf{v}) \}$$

Solution:
- $\mathbf{v}_1$ is eigenvector of $\mathbf{A}$ with largest eigenvalue
- $\mathbf{v}_2$ is eigenvector of $\mathbf{A}$ with smallest eigenvalue
Principal component analysis

Suppose each data point is N-dimensional

- Same procedure applies:

\[
\text{var}(v) = \sum_x \|(x - \bar{x})^T \cdot v\|
\]

\[
= v^T A v \quad \text{where} \quad A = \sum_x (x - \bar{x})(x - \bar{x})^T
\]
Principal component analysis

Suppose each data point is N-dimensional

- Same procedure applies:

\[
\text{var}(v) = \sum_{x} \| (x - \bar{x})^T \cdot v \|
\]

\[
= v^T A v \quad \text{where} \quad A = \sum_{x} (x - \bar{x})(x - \bar{x})^T
\]

- The eigenvectors of \( A \) define a new coordinate system
  - eigenvector with largest eigenvalue captures the most variation among training vectors \( x \)
  - eigenvector with smallest eigenvalue has least variation

\( \bar{x} \) is the mean of the orange points
Principal component analysis

• Compress data by the top few eigenvectors
  – Choosing a “linear subspace”
  – Eigenvectors known as the **principal components**
Implementation issue

• Covariance matrix is huge ($N^2$ for $N$ pixels)

\[ \text{var}(v) = \sum_x \| (x - \bar{x})^T \cdot v \| \]
\[ = v^T A v \quad \text{where} \quad A = \sum_x (x - \bar{x})(x - \bar{x})^T \]

• But typically # examples $<< N$

• Use SVD of $B$
  $A = BB^T$
PCA via Singular Value Decomposition

\[ [u, s, v] = \text{svd}(A); \]

Implementation issue

• Covariance matrix is huge ($N^2$ for $N$ pixels)

\[
\text{var}(v) = \sum_x \| (x - \bar{x})^T \cdot v \|
\]

\[
= v^T A v \quad \text{where} \quad A = \sum_x (x - \bar{x})(x - \bar{x})^T
\]

• But typically # examples $<< N$

• Use SVD of $B$
  \[ A = BB^T \]
  \[ B = USV \]
  \[ BB^T = USV \ V^T S^T \ U^T \]
  \[ = U \ S^2 \ U^T \]
  \[ (BB^T)U = U \ S^2 \]
Eigenfaces (PCA on face images)

1. Compute covariance matrix of face images

2. Compute the principal components ("eigenfaces")
   - K eigenvectors with largest eigenvalues

3. Represent all face images in the dataset as linear combinations of eigenfaces

Eigenfaces (PCA on face images)

1. Compute covariance matrix of face images

2. Compute principal components ("eigenfaces")
   - K eigenvectors with largest eigenvalues

3. Represent all face images in the dataset as linear combinations of eigenfaces

Eigenfaces (PCA on face images)

1. Compute a data matrix of face images

2. Compute SVD ("eigenfaces")
   - (left) $K$ singular vectors with largest singular values

3. Represent all face images in the dataset as linear combinations of eigenfaces

Eigenfaces example

- Training images
- $\mathbf{x}_1, \ldots, \mathbf{x}_N$
Eigenfaces example

Mean: $\mu$

Top eigenvectors: $u_1, \ldots, u_k$
Visualization of eigenfaces

Principal component (eigenvector) $u_k$
Visualization of eigenfaces

Principal component (eigenvector) $u_k$

$\mu + 3\sigma_k u_k$
Visualization of eigenfaces

Principal component (eigenvector) $u_k$

$\mu + 3\sigma_k u_k$

$\mu - 3\sigma_k u_k$
Representation and reconstruction

- Face $\mathbf{x}$ in “face space” coordinates:

$$
\mathbf{x} \rightarrow [u_1^T (\mathbf{x} - \mu), \ldots, u_k^T (\mathbf{x} - \mu)] = w_1, \ldots, w_k
$$
Representation and reconstruction

- Face $x$ in “face space” coordinates:

$$x \rightarrow [u_1^T(x - \mu), \ldots, u_k^T(x - \mu)]$$

$$= \begin{align*}
\hat{x} = & \mu + w_1u_1 + w_2u_2 + w_3u_3 + w_4u_4 + \ldots
\end{align*}$$
Reconstruction

After computing eigenfaces using 400 face images from ORL face database
Reconstruction

After computing eigenfaces using 400 face images from ORL face database
Extrapolating faces

- We can imagine various meaningful directions.
Manipulating faces

• How can we make a face look more female/male, young/old, happy/sad, etc.?

• [http://www.faceresearch.org/demos/transform](http://www.faceresearch.org/demos/transform)

Slide by Kevin Karsch
Assignment 4

1. Computing the mid-way face
2. Morphing
3. Mean face of a population
4. Caricatures