Admin

• A2/Midterm:
  – Grades/solutions posted.
  – Midterms can be viewed during office hours.

• Assignment 4:
  – Due Monday.

• Extra office hours:
  – Thursdays from 4:30-5:30 in ICICS X836.
1. Decision trees
2. Naïve Bayes classification
3. Ordinary least squares regression
4. Logistic regression
5. Support vector machines
6. Ensemble methods
7. Clustering algorithms
8. Principal component analysis
9. Singular value decomposition
10. Independent component analysis
PCA is a linear model for unsupervised learning. Represents features as linear combination of latent factors:

\[ X_{ij} = w_j^T z_i \]

– But we’re learning the latent factors ‘W’ and latent features \( z_i \).

Can also be viewed as an approximate matrix factorization:

\[ X \approx ZW \]
Last Week: Principal Component Analysis (PCA)

- PCA is a linear model for unsupervised learning.
- Represents features as linear combination of latent factors:
  \[ X_{ij} = w_j^T z_i \]
  \[ X_i = W^T z_i \]
- Uses: dimensionality reduction, visualization, factor discovery.

[Diagram of scatter plot with labeled components]

<table>
<thead>
<tr>
<th>Trait</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Openness</td>
<td>Being curious, original, intellectual, creative, and open to new ideas.</td>
</tr>
<tr>
<td>Conscientiousness</td>
<td>Being organized, systematic, punctual, achievement-oriented, and dependable.</td>
</tr>
<tr>
<td>Extraversion</td>
<td>Being outgoing, talkative, sociable, and enjoying social situations.</td>
</tr>
<tr>
<td>Agreeableness</td>
<td>Being affable, tolerant, sensitive, trusting, kind, and warm.</td>
</tr>
<tr>
<td>Neuroticism</td>
<td>Being anxious, irritable, temperamental, and moody.</td>
</tr>
</tbody>
</table>

https://new.edu/resources/big-5-personality-traits
Last Week: Principal Component Analysis (PCA)

- PCA is a linear model for unsupervised learning.
- Represents features as linear combination of latent factors:
  \[ X_{ij} = w_j^T z_i \]
  \[ X_i = W^T z_i \]
- Uses: dimensionality reduction, visualization, factor discovery.

http://www.prismtc.co.uk/superheroes-pca/
Maximizing Variance vs. Minimizing Error

• Our “synthesis” view that PCA minimizes approximation error.
  – Makes connection to k-means and our tricks for linear regression.

• Classic “analysis” view: PCA maximizes variance in $z_i$ space.
  – You pick ‘W’ to explain as much variance in the data as possible.
Choosing Number of Latent Factors

• Common approach to choosing ‘k’:
  – Compute error with k=0: $\| \mathbf{X} \|_F^2 = n \ast \text{var}(x_{ij})$
  – Compare to error with non-zero ‘k’:
    $$\frac{\| Z W - X \|_F^2}{\| X \|_F^2}$$
  – Gives a number between 0 and 1, giving how much “variance remains”.
    • If you want to explain 90% of variance, choose smallest ‘k’ where ratio is < 0.10.
PCA Computation

- The PCA objective with general ‘d’ and ‘k’:
  \[ f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (w_j^T z_i - x_{ij})^2 = \| ZW - X \|_F^2 \]

- 3 common ways to solve this problem:
  - Singular value decomposition (SVD) classic non-iterative approach.
  - Alternating between updating ‘W’ and updating ‘Z’.
  - Stochastic gradient: gradient descent based on random ‘i’ and ‘j’.
    - (Or just plain gradient descent).

- Not convex, all these methods work with random initialization.
The PCA objective with general ‘d’ and ‘k’:

\[
\sum_{j=1}^{d} \sum_{i=1}^{n} (w_j^T z_i - x_{ij})^2 = \|ZW - X\|_F^2
\]

Where we’ve subtracted mean \(\mu_j\) from each feature.

At test time, to find optimal ‘Z’ given ‘W’ for new data:

Given factors ‘W’ and test data \(\hat{X}\):

1. Subtract training mean \(\mu_j\) for each feature \(j\): \(\hat{x}_i = \hat{x}_i - \mu\)
2. Solve for ‘Z’ given ‘W’: \(Z = \hat{X}W^T(WW^T)^{-1}\)

(If \(k=1\) then \(Z_i = \frac{w_c^T x_i}{w_c^T w_c}\))
PCA Non-Uniqueness

• We have the **scaling** problem:

We get some $f(W,Z)$ if you replace '$W$' by $\alpha W$
and '$Z$' by $(\frac{1}{\alpha})Z$ for any $\alpha \neq 0$

$\left(\frac{1}{\alpha}Z\right)(\alpha W) = ZW$

A standard fix: require that $\|w_c\| = 1$ for all factors '$c$'.

Row '$c$' of '$W$'
PCA Non-Uniqueness

• But with multiple PCs, we have new problems:
  – Factors could be non-orthogonal (components interfere with each other):

For $d=2$ and $k=2$

an optimal solution is $W = \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix}$

The standard fix is requiring orthogonal factors: $W_c^T W_c = 0$ when $c \neq c'$

– You can still “rotate” the factors and also have label switching.
  • A fix is to fit the PCs sequentially (can be done with SVD approach):
    1. Find “first” PC $w_c$ that minimizes $\|z w_c^T - x\|_F^2$ (PCA with $k=1$)
    2. Fix “first” PC $w_1$ and find $w_c$ minimizing $\|z w - x\|_F^2$ where $w_1^T w_c = 0$
    3. Fix “first” and “second” PC and find $w_c$ with $w_1^T w_c = 0$ and $w_2^T w_c = 0$
Basis, Orthogonality, Sequential Fitting
Basis, Orthogonality, Sequential Fitting

Any non-parallel line gives optimal solution to second PC (when d=2).

I can get 0 error on every data point.

An optimal solution but not orthogonal. (both PCs give similar information)
Any non-parallel line gives optimal solution to second PC (when d=2).

I can get 0 error on every data point.

An orthogonal solution (PCs are not redundant) but PCs have nothing to do with data.
Basis, Orthogonality, Sequential Fitting

http://setosa.io/ev/principal-component-analysis

optimal solution with one PC ("first" PC)

optimal solution that is orthogonal to "first" PC (sequential fitting of PCs makes directions unique)
Notice that if we require $\|w_c\| = 1$ then $w_c^T w_c = 1$

and if we also require orthogonality $w_c^T w_c' = 0$ for $c \neq c'$ then:

$$WW^T = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_K^T \end{bmatrix} \begin{bmatrix} w_1 & w_2 & \cdots & w_K \end{bmatrix} = \begin{bmatrix} w_1^T w_1 & w_1^T w_2 & \cdots & w_1^T w_K \\ w_2^T w_1 & w_2^T w_2 & \cdots & w_2^T w_K \\ \vdots & \vdots & \ddots & \vdots \\ w_K^T w_1 & w_K^T w_2 & \cdots & w_K^T w_K \end{bmatrix}$$

So finding $Z$ simplifies: $Z = XW^T (WW^T)^\prime = XW^T$

(If $k=1$ then $z_i = w_c^T x_i = w_c^T x_i$)

Do need all this math?
Colour Opponency in the Human Eye

• Classic model of the eye is with 4 photoreceptors:
  – Rods (more sensitive to brightness).
  – L-Cones (most sensitive to red).
  – M-Cones (most sensitive to green).
  – S-Cones (most sensitive to blue).

• Two problems with this system:
  – Correlation between receptors (not orthogonal).
    • Particularly between red/green.
  – We have 4 receptors for 3 colours.

http://oneminuteastronomer.com/astro-course-day-5/
https://en.wikipedia.org/wiki/Color_vision
Colour Opponency in the Human Eye

• Bipolar and ganglion cells seem to code using “opponent colors”:
  – 3-variable orthogonal basis:

• This is similar to PCA (d = 4, k = 3).
Colour Opponency Representation

For this pixel, eye gets 4 signals

\[ w_1 \]
First row of \( W \)
(First PC)

\[ w_2 \]
Second row \((4 \times 1)\)

\[ w_3 \]
Third row \((4 \times 1)\)

Can represent 4 original values with these 3 \( z \) values and matrix \( W \)

= \( W_1 \)

Analogue to means in \( k \)-means.

Brightness

Red/green

Blue/yellow
Application: Face Detection

• Consider problem of face detection:

• Classic methods use “eigenfaces” as basis:
  – PCA applied to images of faces.
Eigenfaces

• Collect a bunch of images of faces under different conditions:

Each row of $X$ will be pixels in one image.

If we have $n$ images that are $m$ by $m$, then $X$ is $n$ by $m^2$. 
Compute mean $\mu_j$ of each column.

Each row of $X$ will be pixels in one image:

Replace each $x_{ij}$ by $x_{ij} - \mu_j$
Compute top \( k \) PCs on centered data:

Each row of \( X \) will be pixels in one image:

\[
X = \begin{bmatrix}
    x_1 - \mu \\
    x_2 - \mu \\
    \vdots \\
    x_n - \mu
\end{bmatrix}
\]
Compute top 'k' PCs on centered data:

"Eigenface" representation:
Eigenfaces

106 of the original faces:

"Eigenface" representation:

\[ x_i = \mu + Z_{i1} \times P_{C1} + Z_{i2} \times P_{C2} + Z_{i3} \times P_{C3} + \ldots \]
Reconstruction with $k=0$

Variance explained: 0%

"Eigenface" representation:

$\mathbf{x}_i = \mu + Z_{i1} \mathbf{PC}_1 + Z_{i2} \mathbf{PC}_2 + Z_{i3} \mathbf{PC}_3 + \ldots$
Eigenfaces

Reconstruction with $k=1$

Variance explained: 34%
Eigenfaces

Reconstruction with $k=2$

PCA Visualization:

"Eigenface" representation:

$xi = \mu + Z_{i1} + Z_{i2} + Z_{i3} + \ldots$

Variance explained: 71%
Reconstruction with $k=3$

Eigenfaces

Variance explained: 76%

PCA Visualization:

"Eigenface" representation:

$$x_i = \mu + Z_{i1} + Z_{i2} + Z_{i3} + \ldots$$

(first row of $W$)
Eigenfaces

Reconstruction with $k=5$

Variance explained: 80%
Eigenfaces

Reconstruction with $k = 10$

Variance explained: 85\%
Reconstruction with $k=21$

Variance explained: 90%
Reconstruction with $k = 54$

Variance explained: 95%
We can replace 10x4 x, values by 54 z, values

Original Images again:

"Eigenfaces"

Plus these mean components and the original images again.
But how should we represent faces?

- **K-means:**
  - ‘Grandmother cell’: one neuron = one face.
  - Almost certainly not true: too few neurons.

- **PCA:**
  - “Distributed representation”.
    - Coded by pattern of group of neurons.
    - Can represent more concepts.
  - But PCA uses positive/negative cancelling parts.

- **Non-negative matrix factorization (NMF):**
  - Latent-factor where $W$ and $Z$ are non-negative.
  - Example of “sparse coding”:
    - Coded by small number of neurons in group.
  - NMF makes object out of of ‘parts’.

Representing Faces

• Why sparse coding?
  – ‘Parts’ are intuitive, and brains seem to use sparse representation.
  – Energy efficiency if using sparse code.
  – Increase number of concepts you can memorize?
    • Some evidence in fruit fly olfactory system.

Summary

• **Analysis view** of PCA is that it maximizes variance.
  – We can choose ‘k’ to explain x% of the variance in the data.
• **Orthogonal basis and sequential fitting** of PCs:
  – Leads to non-redundant PCs with unique directions.
• **Biological motivation** for orthogonal and/or sparse latent factors.
• **Non-negative matrix factorization** leads to sparse LFM.

• Next time: modifying PCA so it splits faces into ‘eyes’, ‘mouths’, etc.