Dimensionality Reduction
&
Principal Component Analysis
Quiz

- Let \( \Sigma \) be the empirical covariance matrix of \( n \) points in \( d \) dimensions

  A. \( \Sigma \) is an \( n \times n \) matrix

  B. \( \Sigma \) is a \( d \times d \) matrix

  C. \( \Sigma \) is a \( m \times m \) matrix where \( m \) is the underlying dimensionality of the \( n \) points (which can be at most \( d \))

  D. \( \text{rank}(\Sigma) \) is \( m \) where \( m \) is the underlying dimensionality of the \( n \) points
We can compress the following images using JPEG?
What if our dataset looked like this?
Eigen Face:

Turk & Pentland’91

Write down each data point as a linear combination of small number of basis vectors.

One of the early successes: in face recognition: classification based on nearest neighbor in the reduced dimension space.
Write down each data point as a linear combination of small number of basis vectors

Turk & Pentland’91

Eigen Face:
Principal Component Analysis (PCA)

Eigen Face:

- Write down each data point as a linear combination of small number of basis vectors
- Data specific compression scheme

Turk & Pentland’91
Principal Component Analysis (PCA)

Eigen Face:

- Write down each data point as a linear combination of small number of basis vectors
- Data specific compression scheme
- One of the early successes: in face recognition: classification based on nearest neighbor in the reduced dimension space

Turk & Pentland’91
How do we represent data?

- Text document: Each coordinate represents a word and the value represents the number of times the word occurred in the document.
- Image: Each coordinate represents a pixel and the value represents the grayscale value of that pixel.
How do we represent data?

Each data-point often represented as vector referred to as feature vector.
Example: Images
Example: Images

vectorize
Example: Images

\[ d = M^2 \]
**Example: Text (Bag of Words)**

*Documents:*

- Car
- Engine
- Hood
- Tires
- Truck
- Trunk

- Car
- Emissions
- Hood
- Make
- Model
- Trunk

- Chomsky
- Corpus
- Noun
- Parsing
- Tagging
- Wonderful
**Example: Text (Bag of Words)**

**Documents:**

- car
- Chomsky
- corpus
- emissions
- engine
- hood
- make
- model
- noun
- parsing
- tagging
- tires
- truck
- trunk
- wonderful

<table>
<thead>
<tr>
<th>car</th>
<th>Chomsky</th>
<th>corpus</th>
<th>emissions</th>
<th>engine</th>
<th>hood</th>
<th>make</th>
<th>model</th>
<th>noun</th>
<th>parsing</th>
<th>tagging</th>
<th>tires</th>
<th>truck</th>
<th>trunk</th>
<th>wonderful</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Given $n$ data points in high-dimensional space, compress them into corresponding $n$ points in lower dimensional space.
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 

DIMENSIONALITY REDUCTION
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K << d$. 

$n \times d$
Given feature vectors \( x_1, \ldots, x_n \in \mathbb{R}^d \), compress the data points into low dimensional representation \( y_1, \ldots, y_n \in \mathbb{R}^K \) where \( K \ll d \).
Given feature vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, compress the data points into low-dimensional representation $y_1, \ldots, y_n \in \mathbb{R}^K$ where $K \ll d$. 

\[
\begin{pmatrix}
X \\
Y
\end{pmatrix}
\rightarrow
\begin{pmatrix}
X \\
Y
\end{pmatrix}^T
\]
Why dimensionality reduction?

- For computational ease
  - As input to supervised learning algorithm
  - Before clustering to remove redundant information and noise
- Data compression & Noise reduction
- Data visualization
Desired properties:

1. Original data can be (approximately) reconstructed
2. Preserve distances between data points
3. "Relevant" information is preserved
4. Noise is reduced
Can we reduce to 1 dim?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95225911</td>
<td>-1.90451821</td>
<td>2.85677732</td>
<td></td>
</tr>
<tr>
<td>0.60681578</td>
<td>-1.21363156</td>
<td>1.82044733</td>
<td></td>
</tr>
<tr>
<td>0.76419773</td>
<td>-1.52839546</td>
<td>2.29259318</td>
<td></td>
</tr>
<tr>
<td>0.44430217</td>
<td>-0.88860435</td>
<td>1.33290652</td>
<td></td>
</tr>
<tr>
<td>0.98425485</td>
<td>-1.9685097</td>
<td>2.95276456</td>
<td></td>
</tr>
<tr>
<td>0.04590113</td>
<td>-0.09180227</td>
<td>0.1377034</td>
<td></td>
</tr>
<tr>
<td>0.52408131</td>
<td>-1.04816263</td>
<td>1.57224394</td>
<td></td>
</tr>
<tr>
<td>0.2887897</td>
<td>-0.5775794</td>
<td>0.8663691</td>
<td></td>
</tr>
<tr>
<td>0.4289135</td>
<td>-0.857827</td>
<td>1.2867405</td>
<td></td>
</tr>
<tr>
<td>0.23877452</td>
<td>-0.47754905</td>
<td>0.71632357</td>
<td></td>
</tr>
<tr>
<td>0.50031855</td>
<td>-1.00063711</td>
<td>1.50095566</td>
<td></td>
</tr>
<tr>
<td>0.7155322</td>
<td>-1.43106441</td>
<td>2.14659661</td>
<td></td>
</tr>
<tr>
<td>0.19638816</td>
<td>-0.39277632</td>
<td>0.58916448</td>
<td></td>
</tr>
<tr>
<td>0.06743744</td>
<td>-0.13487488</td>
<td>0.20231232</td>
<td></td>
</tr>
<tr>
<td>0.18019499</td>
<td>-0.36038997</td>
<td>0.54058496</td>
<td></td>
</tr>
<tr>
<td>0.68941225</td>
<td>-1.37882451</td>
<td>2.06823676</td>
<td></td>
</tr>
<tr>
<td>0.51882043</td>
<td>-1.03764087</td>
<td>1.5564613</td>
<td></td>
</tr>
<tr>
<td>0.71398952</td>
<td>-1.42797904</td>
<td>2.14196857</td>
<td></td>
</tr>
<tr>
<td>0.82712588</td>
<td>-1.30153997</td>
<td>2.53482823</td>
<td></td>
</tr>
</tbody>
</table>
Example:
Students in classroom
Example:
Students in classroom
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information
Dim Reduction: Linear Transformation

1. Pick a low dimensional subspace
2. Project linearly to this subspace
3. Subspace retains as much information as possible

\[ \mathbf{X} \in \mathbb{R}^{n \times d} \]

\[ \mathbf{X}_{\text{new}} \in \mathbb{R}^{n \times d_{\text{new}}} \]
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information

\[ X \times d = W \times K \]
Dim Reduction: Linear Transformation

Pick a low dimensional subspace

Project linearly to this subspace

Subspace retains as much information

\[ n \times d \quad x_{1}^{\top} \quad X \quad x_{n}^{\top} \quad d \]

\[ \times \quad d \quad W \quad = \quad n \quad Y \quad K \]

\[ y_{1}^{\top} \quad y_{n}^{\top} \quad K \]
Dim Reduction: Linear Transformation

Pick a low dimensional subspace
Project linearly to this subspace
Subspace retains as much information

\[ x_1^T \quad x_n^T \]

\[ d \times n \]

\[ x_i^T W \]

\[ y_1^T \quad y_n^T \]

\[ n \times d \]

\[ K \]

\[ y_i^T = x_i^T W \]
Principal Component Analysis (PCA)

Eigen Face:

Write down each data point as a linear combination of small number of basis vectors

Data specific compression scheme

One of the early successes: in face recognition: classification based on nearest neighbor in the reduced dimension space

Turk & Pentland’91
**Eigen Face:**

- Each $x_t$ (each row of $X$) is a face image (vectorized version)
Principal Component Analysis (PCA)

Eigen Face:

- Each $x_t$ (each row of $X$) is a face image (vectorized version)
- Each $y_t$ is the set of coefficients we multiply to the eigen face

Turk & Pentland’91
### Principal Component Analysis (PCA)

**Eigen Face:**

- Each $x_t$ (each row of $X$) is a face image (vectorized version)
- Each $y_t$ is the set of coefficients we multiply to the eigen face
- Each column of $W$ is an Eigenface

Turk & Pentland ’91
Prelude: Reducing to 1 Dim

- W is a $d \times 1$ matrix (d dimensional vector)
- Each data point is compressed to a single number
- How do we pick this $W$?
Prelude: reducing to 1 dimension
Prelude: reducing to 1 dimension
Prelude: reducing to 1 dimension
Prelude: reducing to 1 dimension
Prelude: reducing to 1 dimension

\[ y_1 = w^T x_1 = \|x_1\| \cos (\angle wx_1) \]
Prelude: reducing to 1 dimension

\[ y_1 = w^\top x_1 = \|x_1\| \cos (\angle wx_1) \]

Only direction matters, assume without loss of generality that \(\|w\| = 1\)
PCA: Variance Maximization

Pick directions along which data varies the most

First principal component:

\[ w_1 = \arg \max_{w: w \cdot w = 1} \sum_{t=1}^{n} w \cdot (x_t - \mu) \cdot (x_t - \mu)^T = \arg \max_{w: w \cdot w = 1} w \cdot \Sigma \]

\( \Sigma \) is the covariance matrix.

Writing down Lagrangian and optimizing,

\[ w_1 = \sum_{t=1}^{n} w \cdot \Sigma \cdot w = \]

-1.5 -1 -0.5 0 0.5 1 1.5

-1.2

-1

-0.8

-0.6

-0.4

-0.2

0

0.2

0.4

0.6

0.8

-1.5 -1 -0.5 0 0.5 1 1.5

-1.2

-1

-0.8

-0.6

-0.4

-0.2

0

0.2

0.4

0.6

0.8
PCA: Variance Maximization

Pick directions along which data varies the most

First principal component:

\[ w_1 = \arg \max_{w} \frac{\sum_{t=1}^{n} (w \cdot (x_t - \mu))^2}{\sum_{t=1}^{n} w^2} = \arg \max_{w} \frac{\sum_{t=1}^{n} (w \cdot (x_t - \mu))^2}{\sum_{t=1}^{n} w^2} = \frac{1}{\text{Var}(x)} \]

\[ \text{Var}(x) \]

is the covariance matrix

Writing down Lagrangian and optimizing,

\[ w \cdot \sum_{t=1}^{n} (x_t - \mu)(x_t - \mu)^T = \lambda w \]
PCA: **Variance Maximization**

Pick directions along which data varies the most. The first principal component is:

$$w_1 = \arg \max_{w} \frac{w^T \mathbf{x}_t - \mu}{\|w\|^2} = \arg \max_{w} \frac{w^T \mathbf{x}_t - \mu}{\mathbf{w}^T \mathbf{w}}$$

where $\mathbf{w}$ is the covariance matrix.

Writing down the Lagrangian and optimizing,

$$\mathbf{w} = \mathbf{w}_1$$
Pick directions along which data varies the most
PCA: Variance Maximization

- Pick directions along which data varies the most

\[
\text{Variance} = \frac{1}{n} \sum_{t=1}^{n} \left( y_t - \frac{1}{n} \sum_{s=1}^{n} y_s \right)^2
\]
PCA: Variance Maximization

- Pick directions along which data varies the most

\[
\text{Variance} = \frac{1}{n} \sum_{t=1}^{n} \left( y_t - \frac{1}{n} \sum_{s=1}^{n} y_s \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( w^\top x_t - \frac{1}{n} \sum_{s=1}^{n} w^\top x_s \right)^2
\]
PCA: Variance Maximization

Pick directions along which data varies the most

\[
\text{Variance} = \frac{1}{n} \sum_{t=1}^{n} \left( y_t - \frac{1}{n} \sum_{s=1}^{n} y_s \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{s=1}^{n} \mathbf{w}^\top \mathbf{x}_s \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \mathbf{w}^\top \left( \frac{1}{n} \sum_{s=1}^{n} \mathbf{x}_s \right) \right)^2
\]

\[
\mathbf{w}^\top \text{ is the covariance matrix}
\]
PCA: **Variance Maximization**

- Pick directions along which data varies the most

\[
\text{Variance} = \frac{1}{n} \sum_{t=1}^{n} \left( y_t - \frac{1}{n} \sum_{s=1}^{n} y_s \right)^2 \\
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{s=1}^{n} \mathbf{w}^\top \mathbf{x}_s \right)^2 \\
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \mathbf{w}^\top \left( \frac{1}{n} \sum_{s=1}^{n} \mathbf{x}_s \right) \right)^2 \\
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top (\mathbf{x}_t - \mu) \right)^2
\]
Pick directions along which data varies the most

\[
\text{Variance} = \frac{1}{n} \sum_{t=1}^{n} \left( y_t - \frac{1}{n} \sum_{s=1}^{n} y_s \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{s=1}^{n} \mathbf{w}^\top \mathbf{x}_s \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \mathbf{w}^\top \left( \frac{1}{n} \sum_{s=1}^{n} \mathbf{x}_s \right) \right)^2
\]

\[
= \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top (\mathbf{x}_t - \mu) \right)^2
\]

= average squared inner product
Which Direction?
\[
\frac{1}{n} \sum_{t=1}^{n} (\mathbf{w}^\top (\mathbf{x}_t - \mu))^2 = \frac{1}{n} \sum_{t=1}^{n} \|\mathbf{x}_t - \mu\|^2 \cos(\mathbf{w}, \mathbf{x}_t - \mu)
\]
Which Direction?

\[
\frac{1}{n} \sum_{t=1}^{n} (w^\top (x_t - \mu))^2 = \frac{1}{n} \sum_{t=1}^{n} \|x_t - \mu\|^2 \cos(\langle w, x_t - \mu \rangle)
\]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[
\mathbf{w}_1 = \arg \max_{\mathbf{w} : \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( \mathbf{w}^\top \mathbf{x}_t - \frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top \mathbf{x}_t \right)^2
\]

\[
= \arg \max_{\mathbf{w} : \|\mathbf{w}\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} (\mathbf{w}^\top (\mathbf{x}_t - \mu))^2
\]
Pick directions along which data varies the most
First principal component:

\[ w_1 = \arg \max_{w: \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( w^T x_t - \frac{1}{n} \sum_{t=1}^{n} w^T x_t \right)^2 \]

\[ = \arg \max_{w: \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} (w^T (x_t - \mu))^2 \]

\[ = \arg \max_{w: \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} w^T (x_t - \mu)(x_t - \mu)^T w \]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[ w_1 = \arg \max_{w : \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} \left( w^T x_t - \frac{1}{n} \sum_{t=1}^{n} w^T x_t \right)^2 \]

\[ = \arg \max_{w : \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} (w^T (x_t - \mu))^2 \]

\[ = \arg \max_{w : \|w\|_2 = 1} \frac{1}{n} \sum_{t=1}^{n} w^T (x_t - \mu)(x_t - \mu)^T w \]

\[ = \arg \max_{w : \|w\|_2 = 1} w^T \Sigma w \]

\( \Sigma \) is the covariance matrix
Its a $d \times d$ matrix, $\Sigma[i, j]$ measures "covariance" of features $i$ and $j$

$$\Sigma[i, j] = \frac{1}{n} \sum_{t=1}^{n} (x_t[i] - \mu[i])(x_t[j] - \mu[j])$$
PCA: Variance Maximization

Covariance matrix:

\[ \Sigma = \frac{1}{n} \sum_{t=1}^{n} (\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top \]

- It's a \( d \times d \) matrix, \( \Sigma[i,j] \) measures "covariance" of features \( i \) and \( j \)

\[ \Sigma[i,j] = \frac{1}{n} \sum_{t=1}^{n}(\mathbf{x}_t[i] - \mu[i])(\mathbf{x}_t[j] - \mu[j]) \]
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[ w_1 = \arg \max_{\|w\|_2=1} w^\top \Sigma w \]

\( \Sigma \) is the covariance matrix
PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

\[ w_1 = \arg \max_{w: \|w\|_2=1} w^\top \Sigma w \]

\( \Sigma \) is the covariance matrix

Solution: \( w_1 = \) Largest Eigenvector of \( \Sigma \)
What are Eigen Vectors?
What are Eigen Vectors?

\[ x \mapsto Ax \]
What are Eigen Vectors?

\[ x \mapsto Ax \]

\[ Ax^0 = \lambda x^{0.5} \]

What are Eigen Vectors?

\[ Ax = \lambda x \]

\[ x \mapsto Ax \]
What are Eigen Vectors?

$x \mapsto Ax$

$Ax^0 = \lambda x^0$
Which Direction?

On average parallel
On average orthogonal

Top Eigenvector of covariance matrix
• What if we want more than one number for each data point?

• That is we want to reduce to $K > 1$ dimensions?
PCA: Variance Maximization

How do we find the $K$ components?
How do we find the $K$ components?

Ans: Maximize sum of spread in the $K$ directions
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

This solution is given by $W = \text{Top} K$ eigenvectors of $\mathbf{\Sigma}$. 
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

Find orthonormal $W$ that maximizes
PCA: Variance Maximization

- How do we find the $K$ components?

- We are looking for orthogonal directions that maximize total spread in each direction

- Find orthonormal $W$ that maximizes

$$\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( y_t[j] - \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)^2 = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( w_j^T \left( x_t - \frac{1}{n} \sum_{t=1}^{n} x_t \right) \right)^2$$
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

Find orthonormal $W$ that maximizes

$$
\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( y_t[j] - \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)^2 = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( w_j^\top \left( x_t - \frac{1}{n} \sum_{t=1}^{n} x_t \right) \right)^2
$$

$$
= \sum_{j=1}^{K} w_j^\top \Sigma w_j
$$
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

Find orthonormal $W$ that maximizes

$$
\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( y_t[j] - \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)^2 = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( w_j^T \left( x_t - \frac{1}{n} \sum_{t=1}^{n} x_t \right) \right)^2
$$

$$
= \sum_{j=1}^{K} w_j^T \Sigma w_j
$$
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

Find orthonormal $W$ that maximizes

$$\sum_{j=1}^{K} \frac{1}{n} \left( \sum_{t=1}^{n} y_t[j] - \frac{1}{n} \sum_{t=1}^{n} y_t[j] \right)^2 = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( w_j^T \left( x_t - \frac{1}{n} \sum_{t=1}^{n} x_t \right) \right)^2 = \sum_{j=1}^{K} w_j^T \Sigma w_j$$

This solutions is given by $W = \text{Top } K \text{ eigenvectors of } \Sigma$
How do we find the $K$ components?

We are looking for orthogonal directions that maximize total spread in each direction.

Find orthonormal $W$ that maximizes

$$
\sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} (y_t[j] - \frac{1}{n} \sum_{t=1}^{n} y_t[j])^2 = \sum_{j=1}^{K} \frac{1}{n} \sum_{t=1}^{n} \left( w_j^T \left( x_t - \frac{1}{n} \sum_{t=1}^{n} x_t \right) \right)^2 = \sum_{j=1}^{K} w_j^T \Sigma w_j
$$

Intuition: Remove top direction, now reduce dimension for remaining $d-1$ dimensions.

This solutions is given by $W = \text{Top } K \text{ eigenvectors of } \Sigma$
Eigenvectors of the covariance matrix are the principal components. The top $K$ principal components are the eigenvectors with the $K$ largest eigenvalues.

Projection = Data $\times$ Top $K$ eigenvectors

Reconstruction = Projection $\times$ Transpose of top $K$ eigenvectors

Independently discovered by Pearson in 1901 and Hotelling in 1933.

1. $\Sigma = \text{cov}(X)$

2. $W = \text{eigs}(\Sigma, K)$

3. $Y = X \times W$
Can we reconstruct the original data points?