Do not distribute course material

1) https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch18.pdf

2) http://theory.stanford.edu/~tim/s17/l/l8.pdf

Or you can read the more complete set of notes:
a)http://web.stanford.edu/class/cs168/l/l7.pdf

b) http://theory.stanford.edu/~tim/s17/l/l8.pdf

c) http://web.stanford.edu/class/cs168/l/l9.pdf

Lecture Principal Component Analysis

PROF. LINDA SELLIE

Outline

In this lecture we are not augmenting the feature vector with an extra 1

- ❑ Reduce number of features and find latent features
  - Motivation
  - Intuition on keeping the variance of the data
  - Toy Example of projecting data onto a lower dimensional space
  - Which line should we project onto?
- ❑ Algorithm
- ❑ Examples
- ❑ Finding principle components
- ❑ When PCA doesn't work well

PCA: Principle Component Analysis

Big idea:

1. For each feature, compute the mean. Let $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}$

   (Or zero center the training examples $\mathbf{x}^{(i)}$)

2. Find $k < d$ vectors in $\mathbb{R}^d$: $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \cdots, \mathbf{v}^{(k)}$ which are unit vectors and perpendicular to each other (**orthonormal**)

3. For each training, example compute $\Phi(\mathbf{x}) = (z_1, z_2, \cdots, z_k)$ where $z_i = (\mathbf{x} - \bar{\mathbf{x}})^T\mathbf{v}^{(i)}$

The rest of the lecture is about determining how to choose $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \cdots, \mathbf{v}^{(k)}$ such that $\Phi(\mathbf{x}^{(i)}) = (z_1^{(i)}, z_2^{(i)}, \cdots, z_k^{(i)})$ maximizes variance (and minimizes least-square reconstruction error)

PCA Algorithm

(This algorithm is modified from the book *Learning from Data* by Yasar Abu-Mostafa et al)

- PCA Algorithm:
- Inputs: The centered data matrix X and k >=1
  1) Compute the SVD of X: [U,S,V]=svd (X)
  2) Let V_k = [$\mathbf{v}^{(1)}$,…,$\mathbf{v}^{(k)}$] be the first k columns of V
  3) The PCA-feature matrix is Z = XV_k

k-features for each example

The values we ignore incur the least reconstruction error

$$\hat{X} = (XV_k)V_k^T = ZV_k^T$$

$$Z = \begin{pmatrix} - & \mathbf{x}^{(1)T} & - \\ - & \mathbf{x}^{(2)T} & - \\ & \vdots & \\ - & \mathbf{x}^{(N)T} & - \end{pmatrix} \begin{pmatrix} | & | & & | \\ \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \cdots & \mathbf{v}^{(k)} \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(1)T}\mathbf{v}^{(1)} & \cdots & \mathbf{x}^{(1)T}\mathbf{v}^{(k)} \\ \mathbf{x}^{(2)T}\mathbf{v}^{(1)} & & \mathbf{x}^{(2)T}\mathbf{v}^{(k)} \\ & \vdots & \\ \mathbf{x}^{(N)T}\mathbf{v}^{(1)} & \cdots & \mathbf{x}^{(N)T}\mathbf{v}^{(k)} \end{pmatrix} = \begin{pmatrix} - & \mathbf{z}^{(1)T} & - \\ - & \mathbf{z}^{(2)T} & \\ & \vdots & \\ - & \mathbf{z}^{(N)T} & - \end{pmatrix}$$

More information can be found here: https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

A nice tutorial https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf

We are assuming the diagonal elements in S go from largest to smallest

PCA Algorithm

(This algorithm is modified from the book *Learning from Data* by Yasar Abu-Mostafa et al)

- PCA Algorithm:
- Inputs: The centered data matrix X and k >=1
  1) Compute the SVD of X: [U,S,V]=svd (X)
  2) Let $V_k = [\mathbf{v}^{(1)},\ldots,\mathbf{v}^{(k)}]$ be the first k columns of V
  3) The PCA-feature matrix is $Z = XV_k$

These vectors are the optimal coordinate basis

The values we ignore incur the least reconstruction error

k-features for each example

$$\hat{X} = (XV_k)V_k^T = ZV_k^T$$

If we map back to the origin feature space

$$Z = \begin{pmatrix} - & \boldsymbol{x}^{(1)T} & - \\ - & \boldsymbol{x}^{(2)T} & - \\ & \vdots & \\ - & \boldsymbol{x}^{(N)T} & - \end{pmatrix} \begin{pmatrix} | & | & & | \\ \boldsymbol{v}^{(1)} & \boldsymbol{v}^{(2)} & \cdots & \boldsymbol{v}^{(k)} \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^{(1)T}\boldsymbol{v}^{(1)} & \cdots & \boldsymbol{x}^{(1)T}\boldsymbol{v}^{(k)} \\ \boldsymbol{x}^{(2)T}\boldsymbol{v}^{(1)} & & \boldsymbol{x}^{(2)T}\boldsymbol{v}^{(k)} \\ & \vdots & \\ \boldsymbol{x}^{(N)T}\boldsymbol{v}^{(1)} & \cdots & \boldsymbol{x}^{(N)T}\boldsymbol{v}^{(k)} \end{pmatrix} = \begin{pmatrix} - & \boldsymbol{z}^{(1)T} & - \\ - & \boldsymbol{z}^{(2)T} & \\ & \vdots & \\ - & \boldsymbol{z}^{(N)T} & - \end{pmatrix}$$

More information can be found here: https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

A nice tutorial https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf
We are assuming the diagonal elements in S go from largest to smallest

PCA: Principle Component Analysis

Linearly projects examples into a **lower** dimensional **subspace:** $N \times d$ into $N \times k$ where $k < d$

PCA maintains as much of the original **variance** (and minimizes least square reconstruction error)

Outline

In this lecture we are not augmenting the feature vector with an extra 1

- ❑ Standard **unsupervised** preprocessing: zero centering, data normalization
- ❑ Reduce number of features and find latent features
  - Motivation
  - Intuition on keeping the variance of the data
  - Toy Example of projecting data onto a lower dimensional space
  - Which line should we project onto?
- ❑ Algorithm
- ❑ Examples
- ❑ Finding principle components
- ❑ When PCA doesn't work well

Outline for finding principle components

- What is the "*best*" lower dimensional space?
  - Maximizing variance of the the points projected onto a line **v** (or minimizing mean squared distance between points and their projection onto the line)
  - Goal: find arg $\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X$ and **v** is a unit vector
- How do we find arg $\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X$ and **v** is a unit vector
  - Thought experiment: if D is a diagonal matrix then $\mathbf{v} = \arg\max_v \mathbf{v}^T D \mathbf{v}$ is solved by setting $\mathbf{v} = \mathbf{e}_1$
  - Fact from linear algebra: any $A = X^T X$ can be written as $A = V D V^T$
  - Proving $Ve_1 = \arg\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X = V D V^T$ and v is a unit vector
- Using a standard library to find V
  - Fact from linear algebra: any X can be written as $X = U D V^T$ (singular value decomposition, SVD)
  - There are many libraries to compute the SVD
  - Observe: $X^T X = (U S V^T)^T (U S V^T) = V D V^T$

Intuition regarding the math behind the algorithm

Computing variance of the points

Given the zero-centered data (i.e., *the mean is 0*), $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$:

- The goal is to maximize the variance of the projection of **the training data** onto the vector **v**
- Let **v** be a unit vector $\|v\|_2 = 1$
- The distance of the projected point to the origin is $|\mathbf{x}^T\mathbf{v}|$
- The formula for the variance of a set of N points projected onto the line **v** (the **mean is 0** even after being projected onto **v**)

$$\mathrm{var} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)T}\boldsymbol{v} - 0\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}\right)^2$$

$$\mathbf{x}^{(i)\,T}\mathbf{v} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)})\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix} = \mathbf{v}^T\mathbf{x}^{(i)}$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}\right)\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{v}^T\boldsymbol{x}^{(i)}\right)\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}\right) = \boldsymbol{v}^T\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)T}\right)\boldsymbol{v}$$

maximizing the variance of the projection of **x**

- Maximizing $\mathrm{argmax}_{\mathbf{v}:||\mathbf{v}||=1}\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)T}\mathbf{v})^2$ is the same as $\mathrm{argmax}_{\mathbf{v}:||\mathbf{v}||=1}\mathbf{v}^T X^T X\mathbf{v}$

Computing variance of the points

Given the zero-centered data (i.e., *the mean is 0*), $\mathbf{x}^{(1)},\mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$:

- The goal is to maximize the variance of the projection of **the training data** onto the vector **v**
- Let **v** be a unit vector $||v||_2 = 1$
- The distance of the projected point to the origin is $|\mathbf{x}^T\mathbf{v}|$
- The formula for the variance of a set of N points projected onto the line **v** (the **mean is 0** even after being projected onto **v**)

$$\ldots r = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}-0\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}^{(i)T}\boldsymbol{v}\right)^2$$

$$\mathbf{x}^{(i)\,T}\mathbf{v} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)})\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix} = \mathbf{v}^T\mathbf{x}^{(i)}$$

$$\ldots) = \frac{1}{N}\sum^{N} \ldots \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)T}\Big)\boldsymbol{v}$$

$$\ldots{}^{(i)T}\mathbf{v})^2 \ldots \max_{\mathbf{v}:||\mathbf{v}||=1}\mathbf{v}^TX^TX\mathbf{v}$$

The **v** that maximizes this formula is the first principle component

Same a s X^TX, e.g. if d=2

$$\begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(N)} \\ x_2^{(1)} & x_2^{(2)} & \ldots & x_2^{(N)} \end{bmatrix}\begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \ldots & \\ x_1^{(N)} & x_2^{(N)} \end{bmatrix}$$

Outer product. If d=2

$$\begin{bmatrix} \sum_{i=1}^{N} x_1^{(i)}x_1^{(i)} & \sum_{i=1}^{N} x_1^{(i)}x_2^{(i)} \\ \sum_{i=1}^{N} x_2^{(i)}x_1^{(i)} & \sum_{i=1}^{N} x_2^{(i)}x_2^{(i)} \end{bmatrix}$$

Optimization

$$X = \begin{bmatrix} -\mathbf{x}^{(1)T}- \\ -\mathbf{x}^{(2)T}- \\ \vdots \\ -\mathbf{x}^{(N)T}- \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \vdots \\ v^{(d)} \end{bmatrix}$$

- Assuming X is zero centered
- The ***first principle component*** is v = argmax$_{\mathbf{v}:||\mathbf{v}||=1} \mathbf{v}^T X^T X \mathbf{v}$
- Let $A = X^T X$, our objective is v = argmax$_{\mathbf{v}:||\mathbf{v}||=1} \mathbf{v}^T A \mathbf{v}$

A_{ij} is how frequently the ith and jth features co-occur.
$\frac{1}{N}A$ is called the ***covariance*** matrix

Toy Example

from https://web.stanford.edu/class/cs246/slides/06-dim_red.pdf

$$X=\begin{array}{cc} \begin{array}{ccccc} \text{The Martian} & \text{Interstellar} & \text{Inception} & \text{Titanic} & \text{The Notebook} \end{array} & \\ \begin{array}{l} [1, 1, 1, 0, 0], \\ [3, 3, 3, 0, 0], \\ [4, 4, 4, 0, 0], \\ [5, 5, 5, 0, 0], \\ [0, 2, 0, 4, 4], \\ [0, 0, 0, 5, 5], \\ [0, 1, 0, 2, 2] \end{array} & \begin{array}{l} \text{Billy} \\ \text{Ellie} \\ \text{Sam} \\ \text{Pat} \\ \text{Tully} \\ \text{Liz} \\ \text{Mo} \end{array} \end{array}$$

$$X_{centered}=\begin{bmatrix} -0.86 & -1.29 & -0.86 & -1.57 & -1.57 \\ 1.14 & 0.71 & 1.14 & -1.57 & -1.57 \\ 2.14 & 1.71 & 2.14 & -1.57 & -1.57 \\ 3.14 & 2.71 & 3.14 & -1.57 & -1.57 \\ -1.86 & -0.29 & -1.86 & 2.43 & 2.43 \\ -1.86 & -2.29 & -1.86 & 3.43 & 3.43 \\ -1.86 & -1.29 & -1.86 & 0.43 & 0.43 \end{bmatrix}$$

$$A = X^T_{centered} X_{centered} = \begin{bmatrix} 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ 21.29 & 19.43 & 21.29 & -15.14 & -15.14 \\ 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \end{bmatrix}$$

$$\text{argmax}_{\mathbf{v}:||\mathbf{v}||=1} \mathbf{v}^T \begin{bmatrix} 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ 21.29 & 19.43 & 21.29 & -15.14 & -15.14 \\ 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \end{bmatrix} \mathbf{v}$$

Outline for finding principle components

- What is the *"best"* lower dimensional space?
  - Maximizing variance of the the points projected onto a line v (or minimizing mean squared distance between points and their projection onto the line)
  - Goal: find arg $\max_v v^T A v$, where $A = X^T X$ and v is a unit vector
- How do we find arg $\max_v v^T A v$, where $A = X^T X$ and v is a unit vector
  - Thought experiment: if D is a diagonal matrix then $v = \arg\max_v v^T D v$ is solved by setting $v = e_1$
  - Fact from linear algebra: any $A = X^T X$ can be written as $A = V D V^T$
  - Proving $Ve_1 = \arg\max_v v^T A v$, where $A = X^T X = V D V^T$ and v is a unit vector
- Using a standard library to find V
  - Fact from linear algebra: any X can be written as $X = U D V^T$ (singular value decomposition, SVD)
  - There are many libraries to compute the SVD
  - Observe: $X^T X = (U S V^T)^T (U S V^T) = V D V^T$

An intuitive understanding of PCA

Using the approach from http://theory.stanford.edu/~tim/s17/l/l8.pdf

Simplification

SUPPOSE A IS A DIAGONAL MATRIX

Intuition: Solving the Diagonal Case

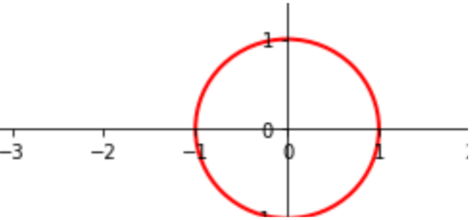
Largest

$$A = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{pmatrix}$$

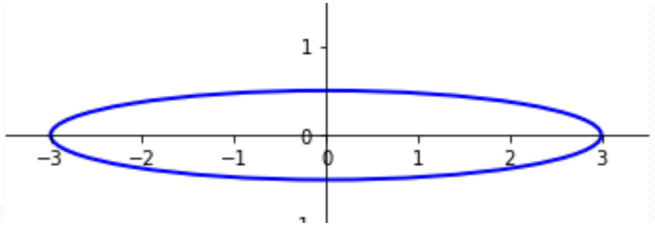
- If A is a diagonal matrix with nonnegative entries $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_d \geq 0$

then the matrix A maps **v** to A**v** where the i^{th} coordinate, v_i, is stretched by a factor of λ_i

- For example if A = $\begin{pmatrix} 3 & 0 \\ 0 & 0.5 \end{pmatrix}$ then A**v** will map v_1 to $3v_1$ and v_2 to $(0.5)v_2$



➜



Easy to see how **v** changes if **v** is on the unit circle

- For the diagonal matrix A, <u>which *unit* vector **v**</u> will be "stretched" the most?

i.e. Which vector will be the longest

$$\mathbf{v}^T(A\mathbf{v}) = (v_1, v_2, \ldots, v_d)\begin{pmatrix} \lambda_1 v_1 \\ \lambda_2 v_2 \\ \vdots \\ \lambda_d v_d \end{pmatrix} = \sum_{i=1}^{d} v_i^2 \lambda_i$$

Weighted average of λ_i

- Since λ_1 is the largest, setting **v** = $\mathbf{e}^{(1)} = (1,0,0,\ldots,0)^T$ maximize $\mathbf{v}^T(A\mathbf{v})$

Using the approach from http://theory.stanford.edu/~tim/s17/l/l8.pdf

Toy Example

$X^T_{centered}\, X_{centered} =$ $A = \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

$$\text{argmax}_{\mathbf{v}:||\mathbf{v}||=1} \mathbf{v}^T \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{v} \;= 110.09 \cdot v_1^2 + 16.73 \cdot v_2^2 + 1.75 \cdot v_3^2 + 0 \cdot v_4^2 + 0 \cdot v_5^2$$

$$\leq 110.09 \cdot (v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2)$$

$$v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2 = 1$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 110.09$$

General Case

$$A = X^T X$$

Rotation/permutation

All a matrix does is rotate/permute, and stretch vectors.

If the columns of V are orthonormal then multiplying a vector **v** by V doesn't change the length of **v**, i.e.

PAIR SHARE: WHAT DOES THIS MEAN? WE COULD ALSO SAY V IS ORTHOGONAL.

Proof: $\|V\mathbf{v}\|_2 = \|\mathbf{v}\|_2$

$$\|V\mathbf{v}\|_2^2 = (V\mathbf{v})^T V\mathbf{v} = \mathbf{v}^T V^T V\mathbf{v} = \mathbf{v}^T\mathbf{v} = \|\mathbf{v}\|_2^2$$

$\|V\mathbf{v}\|_2, \|\mathbf{v}\|_2$ HAVE THE SAME LENGTH (NORM)

Note that: $V^T V = I$

Matrix A can be written: $A = VDV^T$ where D is a diagonal matrix and V is orthogonal.

$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_d \geq 0$

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{pmatrix}$$

What is the direction of "maximum stretch" for A?

Answer: find the vector **v** that is mapped to $\mathbf{e}^{(1)}$ under V^T.

$\mathbf{e}^{(1)} = (1,0,0,\ldots,0)^T$

This is the vector $\mathbf{v} = V\mathbf{e}^{(1)}$.

$\|\mathbf{v}\| = \|V\mathbf{e}^{(1)}\| = 1$

FIRST COLUMN OF V

We can see this by: $\mathbf{v}^T A\mathbf{v} = \mathbf{e}^{(1)T} V^T A V\mathbf{e}^{(1)} = \mathbf{e}^{(1)T} V^T V D V^T V\mathbf{e}^{(1)} = \mathbf{e}^{(1)T} D\mathbf{e}^{(1)} = \lambda_1$

Ideas taken from http://theory.stanford.edu/~tim/s17/l/l8.pdf

Covariance Matrices $\frac{1}{N}A = \frac{1}{N}X^T X$

A fact from linear algebra is that every ***symmetric*** matrix, $A = X^T X$, can be written as

$A = VDV^T$ where V is an orthogonal matrix, and D is a diagonal matrix

If k = 1, the ***first principle component*** **v** is the first row of V^T (i.e.the first column of V, thus we are looking for $\mathbf{v} = V\mathbf{e}^{(1)}$)

$V^{-1}=V^T$

To get the next ***principal component***, repeat this process with $\mathbf{x}^{(i)} := \mathbf{x}^{(i)} - (\mathbf{x}^{(i)T}\mathbf{v})\mathbf{v}$

Remember: $\mathbf{x} = \sum_{i=1}^{d} z_i \mathbf{v}^{(i)} = \sum_{i=1}^{d} (\mathbf{x}^T \mathbf{v}^{(i)})\mathbf{v}^{(i)}$

Observation

Computing the i^{th} principal component

After finding the first principle component we can repeat the procedure on a new data matrix X

$$X = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(N)} \end{bmatrix} \rightarrow X = \begin{bmatrix} \mathbf{x}^{(1)} - (\mathbf{x}^{(1)}\mathbf{v}^{(1)})\mathbf{v}^{(1)} \\ \mathbf{x}^{(2)} - (\mathbf{x}^{(2)}\mathbf{v}^{(1)})\mathbf{v}^{(1)} \\ \vdots \\ \mathbf{x}^{(N)} - (\mathbf{x}^{(N)}\mathbf{v}^{(1)})\mathbf{v}^{(1)} \end{bmatrix}$$

For each example we remove the part of the data that is explained by $\mathbf{v}_1$

By repeating this procedure we can find the first k principle components (i.e. first k eigenvectors)

Toy Example: set **v** to be the first column of V and then compute $\mathbf{v}^T A \mathbf{v}$

$$
X = \begin{array}{c|c}
\text{The Martian, Interstellar, Inception, Titanic, The Notebook} & \\
\hline
[1, 1, 1, 0, 0], & \text{Billy} \\
[3, 3, 3, 0, 0], & \text{Ellie} \\
[4, 4, 4, 0, 0], & \text{Sam} \\
[5, 5, 5, 0, 0], & \text{Pat} \\
[0, 2, 0, 4, 4], & \text{Tully} \\
[0, 0, 0, 5, 5], & \text{Liz} \\
[0, 1, 0, 2, 2] & \text{Mo}
\end{array}
$$

$$
A = X^T_{centered} X_{centered} = \begin{bmatrix} 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ 21.29 & 19.43 & 21.29 & -15.14 & -15.14 \\ 26.86 & 21.29 & 26.86 & -20.43 & -20.43 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \\ -20.43 & -15.14 & -20.43 & 27.71 & 27.71 \end{bmatrix}
$$

$$
A = \begin{bmatrix} -0.47 & -0.36 & -0.39 & -0.71 & 0. \\ -0.37 & -0.41 & 0.83 & -0. & -0. \\ -0.47 & -0.36 & -0.39 & 0.71 & -0. \\ 0.46 & -0.54 & -0.06 & 0. & -0.71 \\ 0.46 & -0.54 & -0.06 & -0. & 0.71 \end{bmatrix} \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.47 & -0.37 & -0.47 & 0.46 & 0.46 \\ -0.36 & -0.41 & -0.36 & -0.54 & -0.54 \\ -0.39 & 0.83 & -0.39 & -0.06 & -0.06 \\ -0.71 & -0. & 0.71 & 0. & 0. \\ 0. & -0. & 0. & -0.71 & 0.71 \end{bmatrix}
$$

$$
\begin{bmatrix} -0.47 & -0.37 & -0.47 & 0.46 & 0.46 \end{bmatrix} \begin{bmatrix} -0.47 & -0.36 & -0.39 & -0.71 & 0. \\ -0.37 & -0.41 & 0.83 & -0. & -0. \\ -0.47 & -0.36 & -0.39 & 0.71 & -0. \\ 0.46 & -0.54 & -0.06 & 0. & -0.71 \\ 0.46 & -0.54 & -0.06 & -0. & 0.71 \end{bmatrix} \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.47 & -0.37 & -0.47 & 0.46 & 0.46 \\ -0.36 & -0.41 & -0.36 & -0.54 & -0.54 \\ -0.39 & 0.83 & -0.39 & -0.06 & -0.06 \\ -0.71 & -0. & 0.71 & 0. & 0. \\ 0. & -0. & 0. & -0.71 & 0.71 \end{bmatrix} \begin{bmatrix} -0.47 \\ -0.37 \\ -0.47 \\ 0.46 \\ 0.46 \end{bmatrix}
$$

$$
\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 110.09
$$

Toy Example

from https://web.stanford.edu/class/cs246/slides/06-dim_red.pdf

$$X = \begin{array}{ccccc|l} \text{The Martian} & \text{Interstellar} & \text{Inception} & \text{Titanic} & \text{The Notebook} & \\ [1, & 1, & 1, & 0, & 0], & \text{Billy} \\ [3, & 3, & 3, & 0, & 0], & \text{Ellie} \\ [4, & 4, & 4, & 0, & 0], & \text{Sam} \\ [5, & 5, & 5, & 0, & 0], & \text{Pat} \\ [0, & 2, & 0, & 4, & 4], & \text{Tully} \\ [0, & 0, & 0, & 5, & 5], & \text{Liz} \\ [0, & 1, & 0, & 2, & 2] & \text{Mo} \end{array}$$

$$X_{\text{centered}} = \begin{bmatrix} -0.86 & -1.29 & -0.86 & -1.57 & -1.57 \\ 1.14 & 0.71 & 1.14 & -1.57 & -1.57 \\ 2.14 & 1.71 & 2.14 & -1.57 & -1.57 \\ 3.14 & 2.71 & 3.14 & -1.57 & -1.57 \\ -1.86 & -0.29 & -1.86 & 2.43 & 2.43 \\ -1.86 & -2.29 & -1.86 & 3.43 & 3.43 \\ -1.86 & -1.29 & -1.86 & 0.43 & 0.43 \end{bmatrix}$$

$$X_{\text{centered}} \; V_2 = \begin{bmatrix} -0.86 & -1.29 & -0.86 & -1.57 & -1.57 \\ 1.14 & 0.71 & 1.14 & -1.57 & -1.57 \\ 2.14 & 1.71 & 2.14 & -1.57 & -1.57 \\ 3.14 & 2.71 & 3.14 & -1.57 & -1.57 \\ -1.86 & -0.29 & -1.86 & 2.43 & 2.43 \\ -1.86 & -2.29 & -1.86 & 3.43 & 3.43 \\ -1.86 & -1.29 & -1.86 & 0.43 & 0.43 \end{bmatrix} \begin{bmatrix} -0.47 & -0.37 \\ -0.36 & -0.41 \\ -0.39 & 0.83 \\ -0.71 & -0. \\ 0. & -0. \end{bmatrix} = \begin{bmatrix} 1.317 & 1.124 \\ 3.95 & 3.372 \\ 5.267 & 4.496 \\ 6.583 & 5.62 \\ -2.9 & 5.121 \\ -4.559 & 5.37 \\ -1.45 & 2.56 \end{bmatrix}$$

Data matrix in Z-space

Toy Example

from https://web.stanford.edu/class/cs246/slides/06-dim_red.pdf

	The Martian	Interstellar	Inception	Titanic	The Notebook	
X=	[1,	1,	1,	0,	0],	Billy
	[3,	3,	3,	0,	0],	Ellie
	[4,	4,	4,	0,	0],	Sam
	[5,	5,	5,	0,	0],	Pat
	[0,	2,	0,	4,	4],	Tully
	[0,	0,	0,	5,	5],	Liz
	[0,	1,	0,	2,	2]	Mo

$$X_{centered}\ V_2 = \begin{bmatrix} -0.86 & -1.29 & -0.86 & -1.57 & -1.57 \\ 1.14 & 0.71 & 1.14 & -1.57 & -1.57 \\ 2.14 & 1.71 & 2.14 & -1.57 & -1.57 \\ 3.14 & 2.71 & 3.14 & -1.57 & -1.57 \\ -1.86 & -0.29 & -1.86 & 2.43 & 2.43 \\ -1.86 & -2.29 & -1.86 & 3.43 & 3.43 \\ -1.86 & -1.29 & -1.86 & 0.43 & 0.43 \end{bmatrix} \begin{bmatrix} -0.47 & -0.37 \\ -0.36 & -0.41 \\ -0.39 & 0.83 \\ -0.71 & -0. \\ 0. & -0. \end{bmatrix} = \begin{bmatrix} 1.317 & 1.124 \\ 3.95 & 3.372 \\ 5.267 & 4.496 \\ 6.583 & 5.62 \\ -2.9 & 5.121 \\ -4.559 & 5.37 \\ -1.45 & 2.56 \end{bmatrix}$$

Data matrix in Z-space

Since $D= \begin{bmatrix} 110.09 & 0 & 0 & 0 & 0 \\ 0 & 16.73 & 0 & 0 & 0 \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

we have maintained $\dfrac{110.09 + 16.73}{110.09 + 16.73 + 1.75 + 0} = \dfrac{126.83}{128.57}$ of the original variance

Toy Example

from https://web.stanford.edu/class/cs246/slides/06-dim_red.pdf

	The Martian	Interstellar	Inception	Titanic	The Notebook	
X= [	1,	1,	1,	0,	0],	Billy
[	3,	3,	3,	0,	0],	Ellie
[	4,	4,	4,	0,	0],	Sam
[	5,	5,	5,	0,	0],	Pat
[	0,	2,	0,	4,	4],	Tully
[	0,	0,	0,	5,	5],	Liz
[	0,	1,	0,	2,	2]	Mo

$$X_{centered}\ V_2 = \begin{bmatrix} -0.86 & -1.29 & -0.86 & -1.57 & -1.57 \\ 1.14 & 0.71 & 1.14 & -1.57 & -1.57 \\ 2.14 & 1.71 & 2.14 & -1.57 & -1.57 \\ 3.14 & 2.71 & 3.14 & -1.57 & -1.57 \\ -1.86 & -0.29 & -1.86 & 2.43 & 2.43 \\ -1.86 & -2.29 & -1.86 & 3.43 & 3.43 \\ -1.86 & -1.29 & -1.86 & 0.43 & 0.43 \end{bmatrix} \begin{bmatrix} -0.47 & -0.37 \\ -0.36 & -0.41 \\ -0.39 & 0.83 \\ -0.71 & -0. \\ 0. & -0. \end{bmatrix} = \begin{bmatrix} 1.317 & 1.124 \\ 3.95 & 3.372 \\ 5.267 & 4.496 \\ 6.583 & 5.62 \\ -2.9 & 5.121 \\ -4.559 & 5.37 \\ -1.45 & 2.56 \end{bmatrix}$$

Data matrix in Z-space

Variance of data explained by $\mathbf{v}^{(1)}$

$$\text{Since } D = \begin{bmatrix} 110.09 & 0 & & & \\ 0 & 16.73 & 0 & & \\ 0 & 0 & 1.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

we have maintained $\dfrac{110.09 + 16.73}{110.09 + 16.73 + 1.75 + 0} = \dfrac{126.83}{128.57}$ of the original variance

Outline for finding principle components

- What is the *"best"* lower dimensional space?
  - Maximizing variance of the the points projected onto a line **v** (or minimizing mean squared distance between points and their projection onto the line)
  - Goal: find arg $\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X$ and v is a unit vector
- How do we find arg $\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X$ and **v** is a unit vector
  - Thought experiment: if D is a diagonal matrix then $\mathbf{v} = \arg\max_v \mathbf{v}^T D \mathbf{v}$ is solved by setting $\mathbf{v} = \mathbf{e}_1$
  - Fact from linear algebra: any $A = X^T X$ can be written as $A = V\ D\ V^T$
  - Proving $V\mathbf{e}_1 = \arg\max_v \mathbf{v}^T A \mathbf{v}$, where $A = X^T X = V\ D\ V^T$ and v is a unit vector
- Using a standard library to find V
  - Fact from linear algebra: any X can be written as $X = U\ S\ V^T$ (singular value decomposition, SVD)
  - There are many libraries to compute the SVD
  - Observe: $X^T X = (U\ S\ V^T)^T (U\ S\ V^T) = V\ D\ V^T$

Computing the PCA using SVD

- Assume X is zero centered
- When computing the PCA, we can directly factorize $A = X^T X = VDV^T$ to find the V_k (eigenvectors) and $\lambda_1, \lambda_2, \ldots, \lambda_k$ (eigenvalues)
- Or we could use singular value decomposition (SVD) of X to find V_k (eigenvectors) and $\lambda_1, \lambda_2, \ldots, \lambda_k$ (eigenvalues). Using this method, we don't have to compute X^TX, which can be extremely large if there are many features. We can see by the following calculations that we get the same answer:

By **SVD**, we factorize $X = U S V^T$.

For X is zero-centered, then $A = X^TX = (U S V^T)^T(U S V^T) = (V S^T U^T)(U S V^T)$.

Note that $U^TU = I$ (identity matrix). Since S is a diagonal matrix, then $S^T = S$. Let $D = S^TS = S^2$

Thus $A = X^TX = (V S U^T)(U S V^T) = VS^2V^T = VDV^T$.

Outline

In this lecture we are not augmenting the feature vector with an extra 1

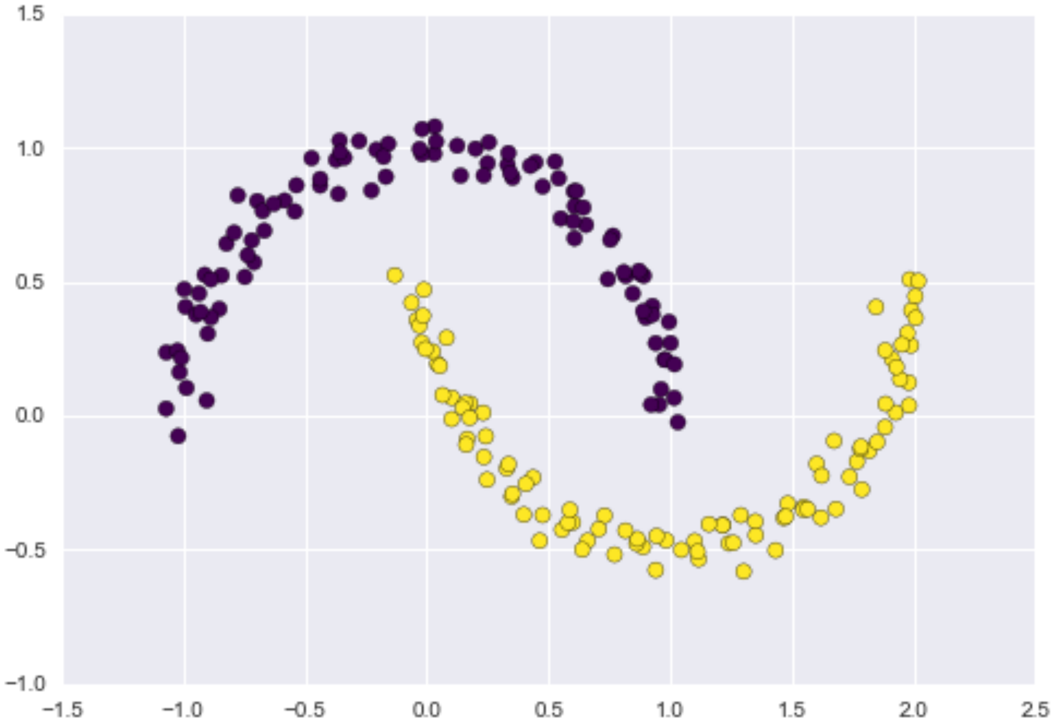
- ❑ Standard **unsupervised** preprocessing: zero centering, data normalization
- ❑ Reduce number of features and find latent features
  - Motivation
  - Intuition on keeping the variance of the data
  - Toy Example of projecting data onto a lower dimensional space
  - Which line should we project onto?
- ❑ Algorithm
- ❑ Examples
- ❑ Finding principle components
- ❑ When PCA doesn't work well

When PCA Doesn't Work Well

- PCA doesn't find non-linear features

- One option is to transform the features before applying PCA

- To learn about kernels with PCA you can look at: https://en.m.wikipedia.org/wiki/Kernel_principal_component_analysis



Summary

- PCA is a linear projection of our original feature space into a k dimensional vector space
- The new feature space is in a new coordinate system
- We choose the new coordinates such that:
  - maximizes variance
  - minimizes projection error (square loss)
- PCA is used for:
  - dimensionality reduction
  - visualization (if we choose k to be 2 or 3)
  - Compression (with loss)
  - De-noising (removes small variance in data)

Additional notes

- The *data must be centered* before running PCA
- Optional: normalize none, some, or all the features
  - Normalizing will prevent arbitrary scales of different features affect the decisions made by the PCA algorithm
  - Not normalizing when features have similar scales with different variances will keep differences of features that are usually meaningful
- Choosing k (i.e. size of subspace $\mathbb{R}^k$ for new features)
  - If performing PCA as a preprocessing step before running a learning algorithm, choose k based on validation data performance
  - Optionally: choose k based on how much of the original variance of X is maintained after projecting to a smaller dimension is $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$. Note: X is noisy so this formula includes the noise.
  - Learn more at https://ro-che.info/articles/2017-12-11-pca-explained-variance

Additional notes

- The *data must be centered* before running PCA
- Optional: normalize none, some, or all the features
  - Normalizing will prevent arbitrary scales of different features affect the decisions made by the PCA algorithm
  - Not normalizing when features have similar scales with different variances will keep differences of features that are usually meaningful
- Choosing k (i.e. size of subspace $\mathbb{R}^k$ for new features)
  - If performing PCA as a preprocessing step before running a learning algorithm, choose k based on validation data performance
  - Optionally: choose k based on how much of the original variance of X is maintained after proj… ion is $\dfrac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$. Note: X is noisy so this formula includes the noise.

    The $\lambda_1, \lambda_2, \cdots, \lambda_d$ are the diagonal values in D = S^2
  - Learn more at https://ro-che.info/articles/2017-12-11-pca-explained-variance