

DSC 140B

Representation Learning

Lecture 10 | Part 1

Covariance Matrices

Variance

- ▶ We know how to compute the variance of a set of numbers $X = \{x^{(1)}, \dots, x^{(n)}\}$:

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (x^{(i)} - \mu)^2$$

- ▶ The variance measures the “spread” of the data

Generalizing Variance

- ▶ If we have two features, x_1 and x_2 , we can compute the variance of each as usual:

$$\text{Var}(x_1) = \frac{1}{n} \sum_{i=1}^n (\vec{X}_1^{(i)} - \mu_1)^2$$

$$\text{Var}(x_2) = \frac{1}{n} \sum_{i=1}^n (\vec{X}_2^{(i)} - \mu_2)^2$$

- ▶ Can also measure how x_1 and x_2 vary together.

Measuring Similar Information

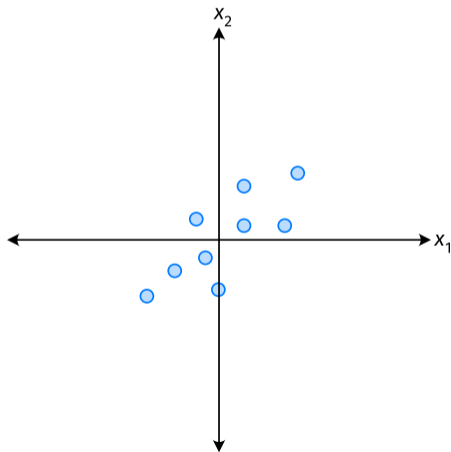
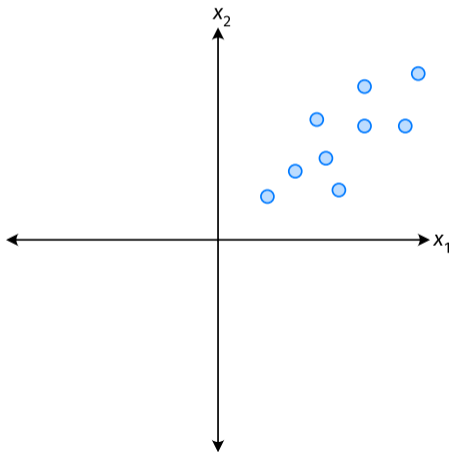
- ▶ Features which share information if they *vary together*.
 - ▶ A.k.a., they “co-vary”
- ▶ Positive association: when one is above average, so is the other
- ▶ Negative association: when one is above average, the other is below average

Examples

- ▶ Positive: temperature and ice cream cones sold.
- ▶ Positive: temperature and shark attacks.
- ▶ Negative: temperature and coats sold.

Centering

- First, it will be useful to **center** the data.



Centering

- ▶ Compute the mean of each feature:

$$\mu_j = \frac{1}{n} \sum_1^n \vec{x}_j^{(i)}$$

- ▶ Define new centered data:

$$\vec{z}^{(i)} = \begin{pmatrix} \vec{x}_1^{(i)} - \mu_1 \\ \vec{x}_2^{(i)} - \mu_2 \\ \vdots \\ \vec{x}_d^{(i)} - \mu_d \end{pmatrix}$$

Centering (Equivalently)

- ▶ Compute the mean of all data points:

$$\mu = \frac{1}{n} \sum_1^n \vec{x}^{(i)}$$

- ▶ Define new centered data:

$$\vec{z}^{(i)} = \vec{x}^{(i)} - \mu$$

Exercise

Center the data set:

$$\vec{x}^{(1)} = (1, 2, 3)^T$$

$$\vec{x}^{(2)} = (-1, -1, 0)^T$$

$$\vec{x}^{(3)} = (0, 2, 3)^T$$

Quantifying Co-Variance

- ▶ One approach is as follows¹.

$$\text{Cov}(x_i, x_j) = \frac{1}{n} \sum_{k=1}^n \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

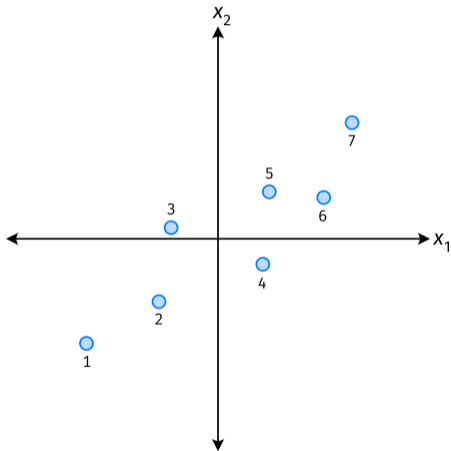
- ▶ For each data point, multiply the value of feature i and feature j , then average these products.
- ▶ This is the **covariance** of features i and j .

¹Assuming centered data

Quantifying Covariance

- ▶ Assume the data are **centered**.

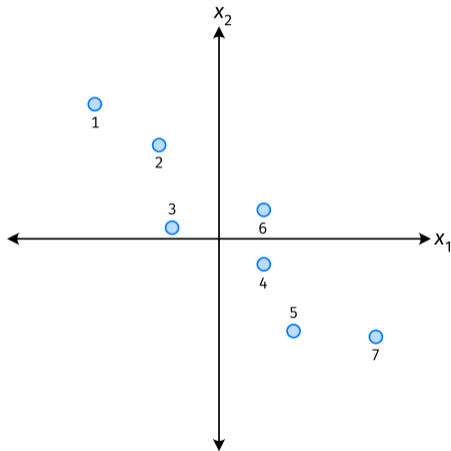
$$\text{Covariance} = \frac{1}{7} \sum_{i=1}^7 \vec{X}_1^{(i)} \times \vec{X}_2^{(i)}$$



Quantifying Covariance

- ▶ Assume the data are **centered**.

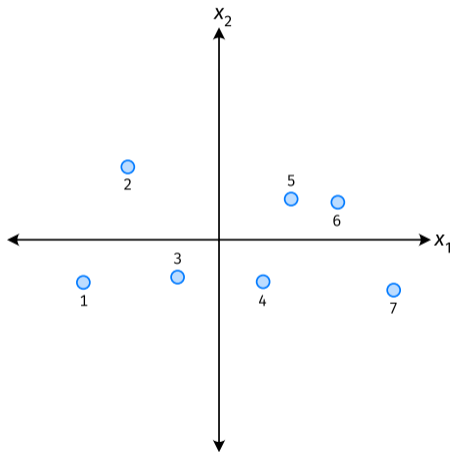
$$\text{Covariance} = \frac{1}{7} \sum_{i=1}^7 \vec{X}_1^{(i)} \times \vec{X}_2^{(i)}$$



Quantifying Covariance

- ▶ Assume the data are **centered**.

$$\text{Covariance} = \frac{1}{7} \sum_{i=1}^7 \vec{X}_1^{(i)} \times \vec{X}_2^{(i)}$$



Quantifying Covariance

- ▶ The **covariance** quantifies extent to which two variables vary together.
- ▶ Assume we have centered the data.
- ▶ The **sample covariance** of feature i and j is:

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^n \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Exercise

True or False: $\sigma_{ij} = \sigma_{ji}$?

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^n \vec{X}_i^{(k)} \vec{X}_j^{(k)}$$

Covariance Matrices

- ▶ Given data $\vec{x}^{(1)}, \dots, \vec{x}^{(n)} \in \mathbb{R}^d$.
- ▶ The **sample covariance matrix** C is the $d \times d$ matrix whose ij entry is defined to be σ_{ij} .

$$\sigma_{ij} = \frac{1}{n} \sum_{k=1}^n \vec{x}_i^{(k)} \vec{x}_j^{(k)}$$

Observations

- ▶ Diagonal entries of C are the variances.
- ▶ The matrix is **symmetric!**

Note

- ▶ Sometimes you'll see the sample covariance defined as:

$$\sigma_{ij} = \frac{1}{n-1} \sum_{k=1}^n \vec{X}_i^{(k)} \vec{X}_j^{(k)}$$

Note the $1/(n-1)$

- ▶ This is an **unbiased** estimator of the population covariance.
- ▶ Our definition is the **maximum likelihood** estimator.
- ▶ In practice, it doesn't matter: $1/(n-1) \approx 1/n$.
- ▶ For consistency, in this class use $1/n$.

Computing Covariance

- ▶ There is a “trick” for computing sample covariance matrices.
- ▶ Step 1: make $n \times d$ data matrix, X
- ▶ Step 2: make Z by centering columns of X
- ▶ Step 3: $C = \frac{1}{n}Z^T Z$

Computing Covariance (in code)²

```
»» mu = X.mean(axis=0)
»» Z = X - mu
»» C = 1 / len(X) * Z.T @ Z
```

²Or use `np.cov`

DSC 140B

Representation Learning

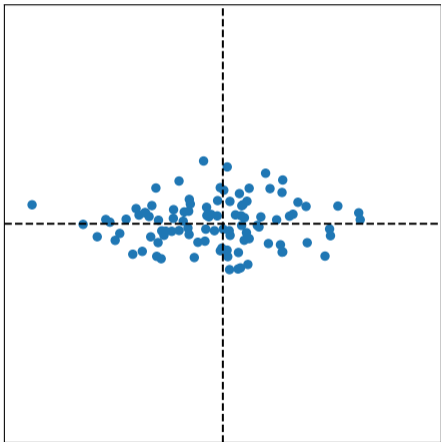
Lecture 10 | Part 2

Visualizing Covariance Matrices

Visualizing Covariance Matrices

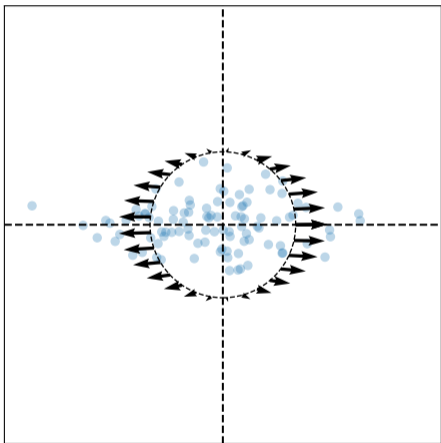
- ▶ Covariance matrices are symmetric.
- ▶ They have axes of symmetry (eigenvectors and eigenvalues).
- ▶ What are they?

Visualizing Covariance Matrices



$$C \approx \begin{pmatrix} & \\ & \end{pmatrix}$$

Visualizing Covariance Matrices

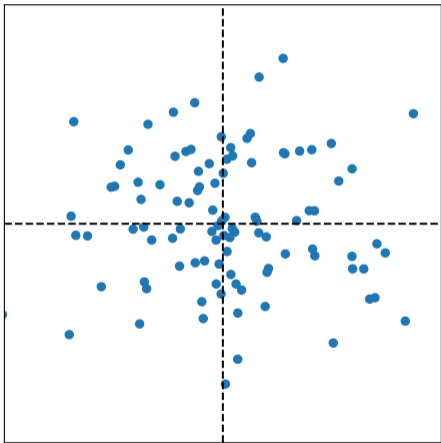


Eigenvectors:

$$\vec{u}^{(1)} \approx$$

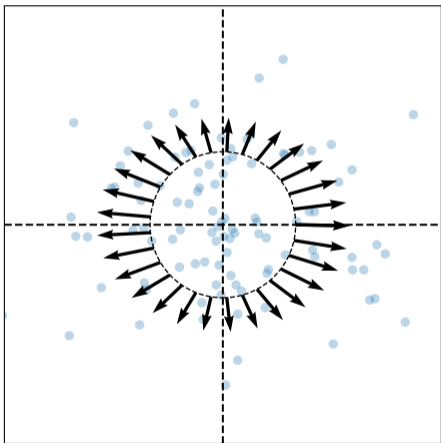
$$\vec{u}^{(2)} \approx$$

Visualizing Covariance Matrices



$$C \approx \begin{pmatrix} & \\ & \end{pmatrix}$$

Visualizing Covariance Matrices

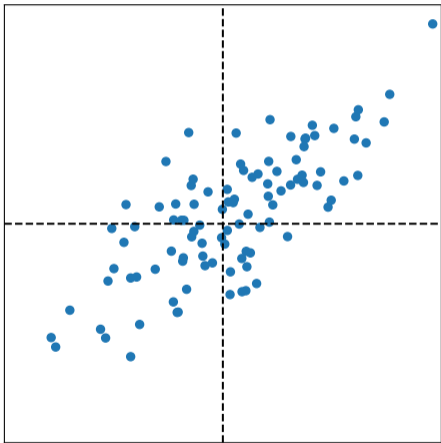


Eigenvectors:

$$\vec{u}^{(1)} \approx$$

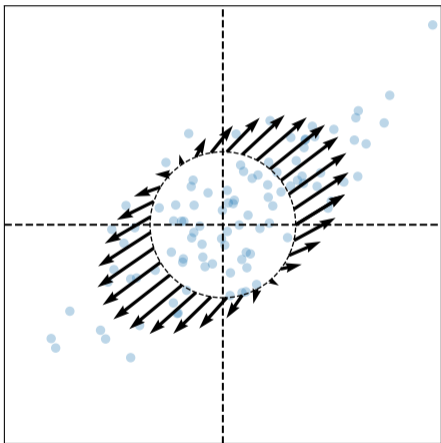
$$\vec{u}^{(2)} \approx$$

Visualizing Covariance Matrices



$$C \approx \begin{pmatrix} & \\ & \end{pmatrix}$$

Visualizing Covariance Matrices



Eigenvectors:

$$\vec{u}^{(1)} \approx$$

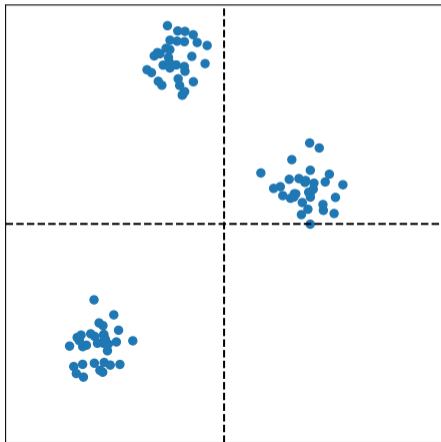
$$\vec{u}^{(2)} \approx$$

Intuitions

- ▶ The **eigenvectors** of the covariance matrix describe the data's "principal directions"
 - ▶ C tells us something about data's shape.
- ▶ The **top eigenvector** points in the direction of "maximum variance".
- ▶ The **top eigenvalue** is proportional to the variance in this direction.

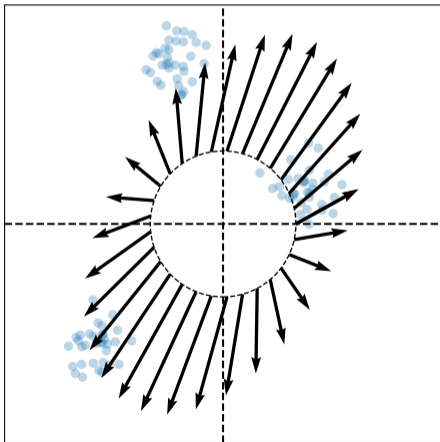
Caution

- ▶ The data doesn't always look like this.
- ▶ We can always compute covariance matrices.
- ▶ They just may not describe the data's shape very well.



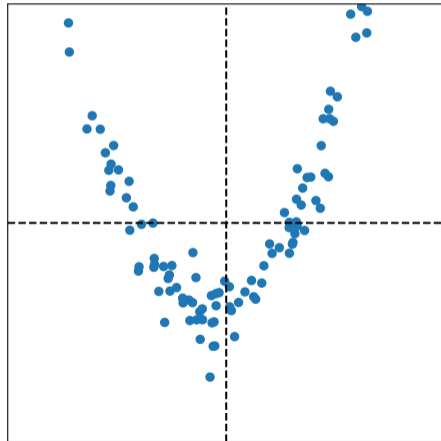
Caution

- ▶ The data doesn't always look like this.
- ▶ We can always compute covariance matrices.
- ▶ They just may not describe the data's shape very well.



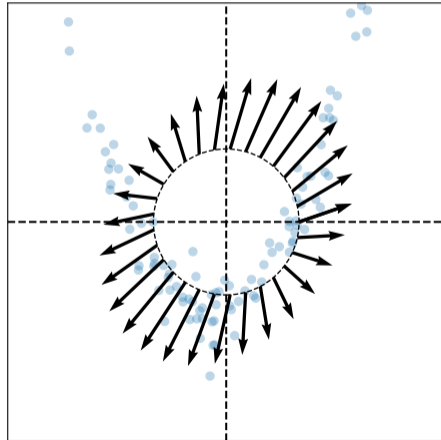
Caution

- ▶ The data doesn't always look like this.
- ▶ We can always compute covariance matrices.
- ▶ They just may not describe the data's shape very well.



Caution

- ▶ The data doesn't always look like this.
- ▶ We can always compute covariance matrices.
- ▶ They just may not describe the data's shape very well.



DSC 140B

Representation Learning

Lecture 10 | Part 3

PCA, More Formally

The Story (So Far)

- ▶ We want to create a single new feature, z .
- ▶ Our idea: $z = \vec{x} \cdot \vec{u}$; choose \vec{u} to point in the “direction of maximum variance”.
- ▶ Intuition: the top eigenvector of the covariance matrix points in direction of maximum variance.

More Formally...

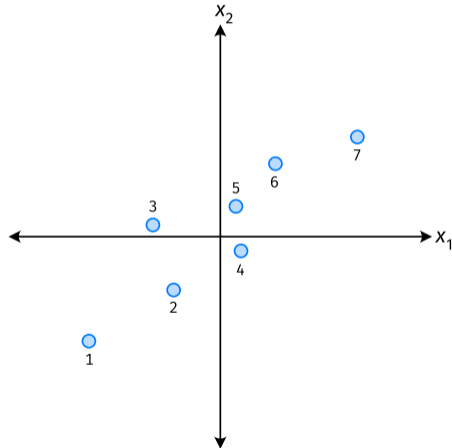
- ▶ We haven't actually defined "direction of maximum variance"
- ▶ Let's derive PCA more formally.

Variance in a Direction

- ▶ Let \vec{u} be a unit vector.
- ▶ $z^{(i)} = \vec{x}^{(i)} \cdot \vec{u}$ is the new feature for $\vec{x}^{(i)}$.
- ▶ The variance of the new features is:

$$\begin{aligned}\text{Var}(z) &= \frac{1}{n} \sum_{i=1}^n (z^{(i)} - \mu_z)^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\vec{x}^{(i)} \cdot \vec{u} - \mu_z)^2\end{aligned}$$

Example



Note

- ▶ If the data are centered, then $\mu_z = 0$ and the variance of the new features is:

$$\begin{aligned}\text{Var}(z) &= \frac{1}{n} \sum_{i=1}^n (z^{(i)})^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\vec{x}^{(i)} \cdot \vec{u})^2\end{aligned}$$

Goal

- ▶ The variance of a data set in the direction of \vec{u} is:

$$g(\vec{u}) = \frac{1}{n} \sum_{i=1}^n (\vec{x}^{(i)} \cdot \vec{u})^2$$

- ▶ Our goal: Find a unit vector \vec{u} which maximizes g .

Claim

$$\frac{1}{n} \sum_{i=1}^n (\vec{x}^{(i)} \cdot \vec{u})^2 = \vec{u}^T C \vec{u}$$

Our Goal (Again)

- ▶ Find a unit vector \vec{u} which maximizes $\vec{u}^T C \vec{u}$.

Claim

- ▶ To maximize $\vec{u}^T C \vec{u}$ over unit vectors, choose \vec{u} to be the top eigenvector of C .
- ▶ Proof:

PCA (for a single new feature)

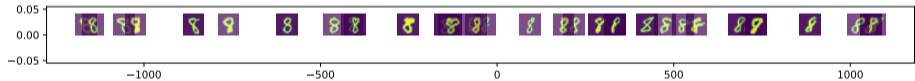
- **Given:** data points $\vec{x}^{(1)}, \dots, \vec{x}^{(n)} \in \mathbb{R}^d$
1. Compute the covariance matrix, C .
 2. Compute the top eigenvector \vec{u} , of C .
 3. For $i \in \{1, \dots, n\}$, create new feature:

$$z^{(i)} = \vec{u} \cdot \vec{x}^{(i)}$$

A Parting Example

- ▶ MNIST: 60,000 images in 784 dimensions
- ▶ Principal component: $\vec{u} \in \mathbb{R}^{784}$
- ▶ We can project an image in \mathbb{R}^{784} onto \vec{u} to get a single number representing the image

Example



DSC 140B

Representation Learning

Lecture 10 | Part 4

Dimensionality Reduction with $d \geq 2$

So far: PCA

- ▶ **Given:** data $\vec{x}^{(1)}, \dots, \vec{x}^{(n)} \in \mathbb{R}^d$
- ▶ **Map:** each data point $\vec{x}^{(i)}$ to a single feature, z_i .
 - ▶ Idea: maximize the variance of the new feature
- ▶ **PCA:** Let $z_i = \vec{x}^{(i)} \cdot \vec{u}$, where \vec{u} is top eigenvector of covariance matrix, C .

Now: More PCA

- ▶ **Given:** data $\vec{x}^{(1)}, \dots, \vec{x}^{(n)} \in \mathbb{R}^d$
- ▶ **Map:** each data point $\vec{x}^{(i)}$ to k new features, $\vec{z}^{(i)} = (z_1^{(i)}, \dots, z_k^{(i)})$.

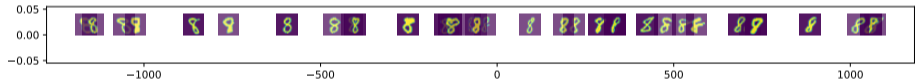
A Single Principal Component

- ▶ Recall: the **principal component** is the top eigenvector \vec{u} of the covariance matrix, C
- ▶ It is a unit vector in \mathbb{R}^d
- ▶ Make a new feature $z \in \mathbb{R}$ for point $\vec{x} \in \mathbb{R}^d$ by computing $z = \vec{x} \cdot \vec{u}$
- ▶ This is dimensionality reduction from $\mathbb{R}^d \rightarrow \mathbb{R}^1$

Example

- ▶ MNIST: 60,000 images in 784 dimensions
- ▶ Principal component: $\vec{u} \in \mathbb{R}^{784}$
- ▶ We can project an image in \mathbb{R}^{784} onto \vec{u} to get a single number representing the image

Example



Another Feature?

- ▶ Clearly, mapping from $\mathbb{R}^{784} \rightarrow \mathbb{R}^1$ loses a lot of information
- ▶ What about mapping from $\mathbb{R}^{784} \rightarrow \mathbb{R}^2? \mathbb{R}^k?$

A Second Feature

- ▶ Our first feature is a mixture of features, with weights given by unit vector $\vec{u}^{(1)} = (u_1^{(1)}, u_2^{(1)}, \dots, u_d^{(1)})^T$.

$$z_1 = \vec{u}^{(1)} \cdot \vec{x} = u_1^{(1)}x_1 + \dots + u_d^{(1)}x_d$$

- ▶ To maximize variance, choose $\vec{u}^{(1)}$ to be top eigenvector of C .

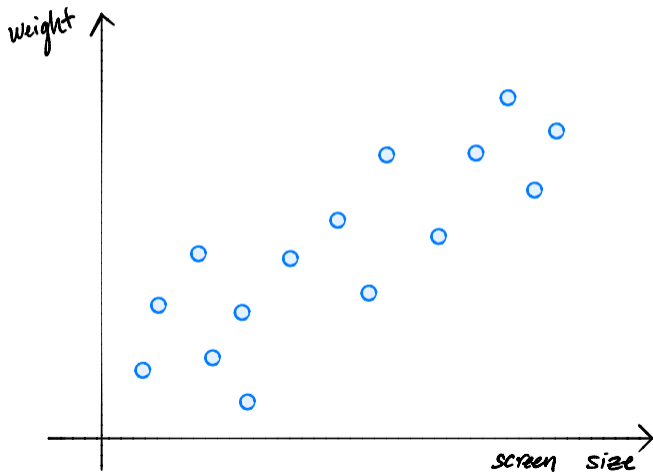
A Second Feature

- ▶ Make same assumption for second feature:

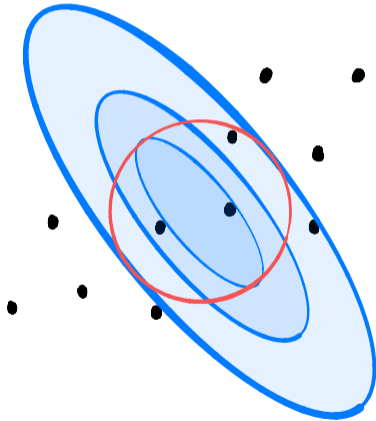
$$z_2 = \vec{u}^{(2)} \cdot \vec{x} = u_1^{(2)}x_1 + \dots + u_d^{(2)}x_d$$

- ▶ How do we choose $\vec{u}^{(2)}$?
- ▶ We should choose $\vec{u}^{(2)}$ to be **orthogonal** to $\vec{u}^{(1)}$.
 - ▶ No “redundancy”.

A Second Feature



A Second Feature



Intuition

- ▶ Claim: if \vec{u} and \vec{v} are eigenvectors of a symmetric matrix with distinct eigenvalues, they are orthogonal.
- ▶ We should choose $\vec{u}^{(2)}$ to be an **eigenvector** of the covariance matrix, C .
- ▶ The second eigenvector of C is called the **second principal component**.

A Second Principal Component

- ▶ Given a covariance matrix C .
- ▶ The principal component $\vec{u}^{(1)}$ is the top eigenvector of C .
 - ▶ Points in the direction of maximum variance.
- ▶ The *second* principal component $\vec{u}^{(2)}$ is the *second* eigenvector of C .
 - ▶ Out of all vectors orthogonal to the principal component, points in the direction of max variance.

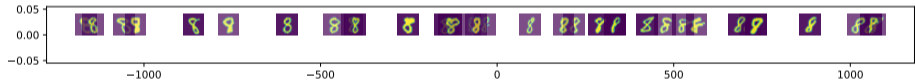
PCA: Two Components

- ▶ Given data $\{\vec{x}^{(1)}, \dots, \vec{x}^{(n)}\} \in \mathbb{R}^d$.
- ▶ Compute covariance matrix C , top two eigenvectors $\vec{u}^{(1)}$ and $\vec{u}^{(2)}$.
- ▶ For any vector $\vec{x} \in \mathbb{R}^d$, its new representation in \mathbb{R}^2 is $\vec{z} = (z_1, z_2)^T$, where:

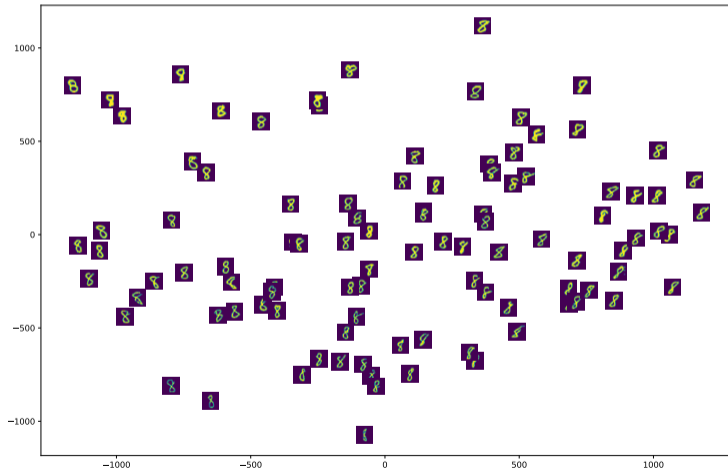
$$z_1 = \vec{x} \cdot \vec{u}^{(1)}$$

$$z_2 = \vec{x} \cdot \vec{u}^{(2)}$$

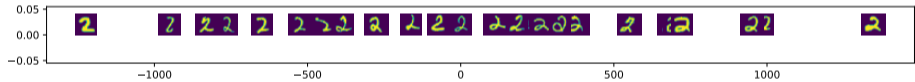
Example



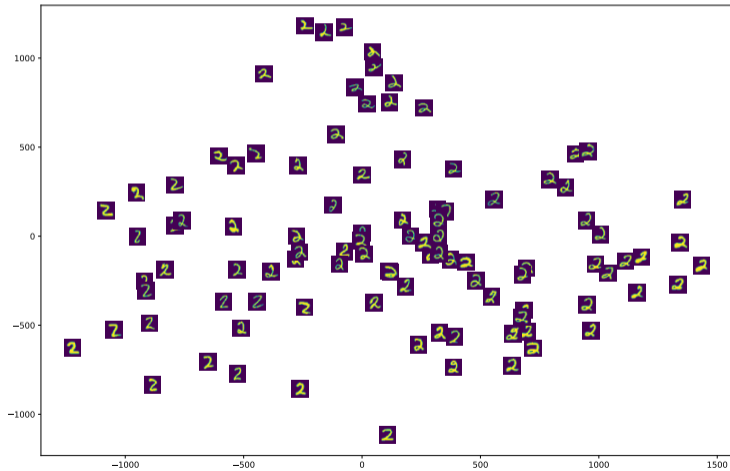
Example



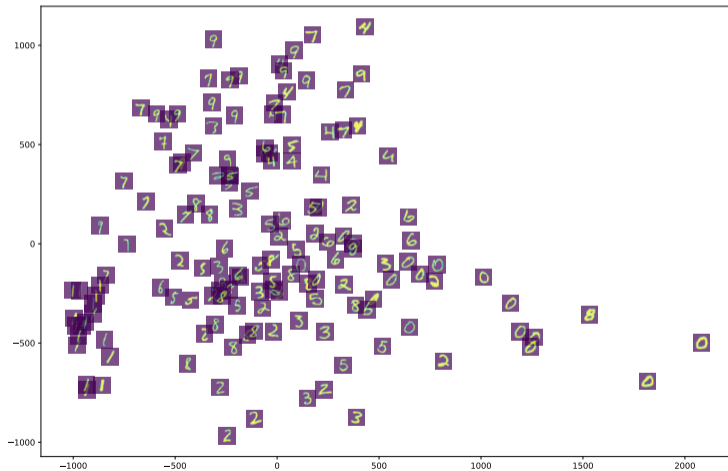
Example



Example



Example



PCA: k Components

- ▶ Given data $\{\vec{x}^{(1)}, \dots, \vec{x}^{(n)}\} \in \mathbb{R}^d$, number of components k .
- ▶ Compute covariance matrix C , top $k \leq d$ eigenvectors $\vec{u}^{(1)}, \vec{u}^{(2)}, \dots, \vec{u}^{(k)}$.
- ▶ For any vector $\vec{x} \in \mathbb{R}^d$, its new representation in \mathbb{R}^k is $\vec{z} = (z_1, z_2, \dots, z_k)^T$, where:

$$z_1 = \vec{x} \cdot \vec{u}^{(1)}$$

$$z_2 = \vec{x} \cdot \vec{u}^{(2)}$$

$$\vdots$$

$$z_k = \vec{x} \cdot \vec{u}^{(k)}$$

Matrix Formulation

- ▶ Let X be the **data matrix** (n rows, d columns)
- ▶ Let U be matrix of the k eigenvectors as columns (d rows, k columns)
- ▶ The new representation: $Z = XU$

DSC 140B

Representation Learning

Lecture 10 | Part 5

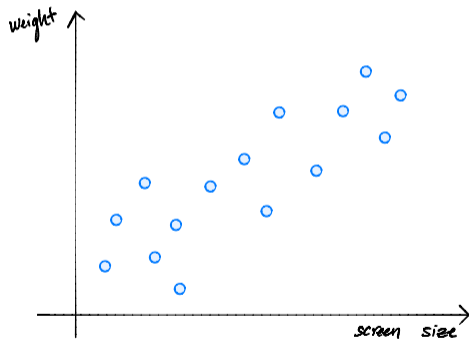
Reconstructions

Reconstructing Points

- ▶ PCA helps us reduce dimensionality from $\mathbb{R}^d \rightarrow \mathbb{R}^k$
- ▶ Suppose we have the “new” representation in \mathbb{R}^k .
- ▶ Can we “go back” to \mathbb{R}^d ?
- ▶ And why would we want to?

Back to \mathbb{R}^d

- ▶ Suppose new representation of \vec{x} is z .
- ▶ $z = \vec{x} \cdot \vec{u}^{(1)}$
- ▶ Idea: $\vec{x} \approx z\vec{u}^{(1)}$



Reconstructions

- ▶ Given a “new” representation of \vec{x} , $\vec{z} = (z_1, \dots, z_k) \in \mathbb{R}^k$
- ▶ And top k eigenvectors, $\vec{u}^{(1)}, \dots, \vec{u}^{(k)}$
- ▶ The **reconstruction** of \vec{x} is

$$z_1 \vec{u}^{(1)} + z_2 \vec{u}^{(2)} + \dots + z_k \vec{u}^{(k)} = U \vec{z}$$

Reconstruction Error

- ▶ The reconstruction *approximates* the original point, \vec{x} .
- ▶ The **reconstruction error** for a single point, \vec{x} :

$$\|\vec{x} - U\vec{z}\|^2$$

- ▶ Total reconstruction error:

$$\sum_{i=1}^n \|\vec{x}^{(i)} - U\vec{z}^{(i)}\|^2$$

