

DSC 190

Machine Learning: Representations

Lecture 9 | Part 1

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...

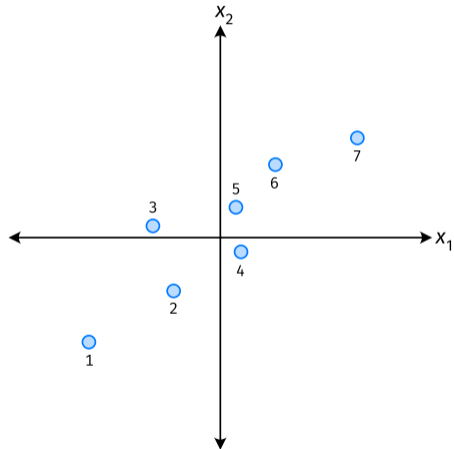
- ▶ We haven't actually defined "direction of maximum variance"
- ▶ Let's derive PCA more formally.

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:

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

DSC 190

Machine Learning: Representations

Lecture 9 | Part 2

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 .

Today: 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,
 $\vec{z}^{(i)} = (z_1^{(i)}, \dots, z_k^{(i)})$.

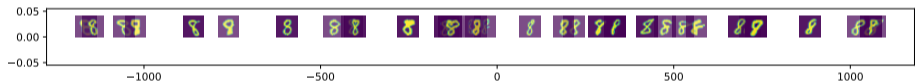
A Single Principal Component

- ▶ 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 .

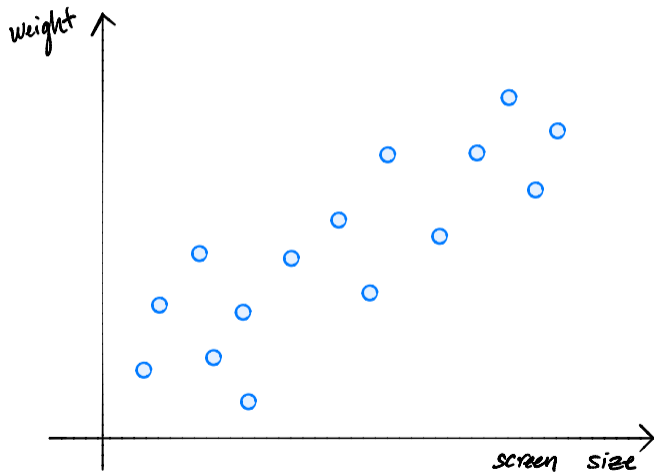
A Second Feature

- ▶ Make same assumption for second feature:

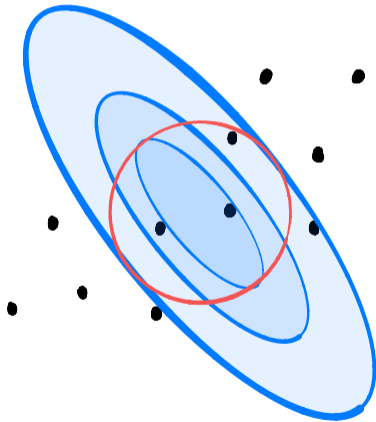
$$z_2 = \vec{u}^{(2)} \cdot \vec{x} = u_1^{(2)} x_1 + \dots + u_d^{(2)} x_d$$

- ▶ 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.

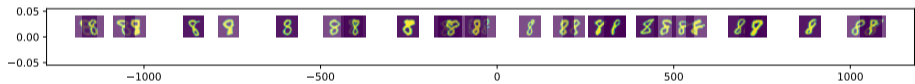
PCA: Two Components

- ▶ 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:

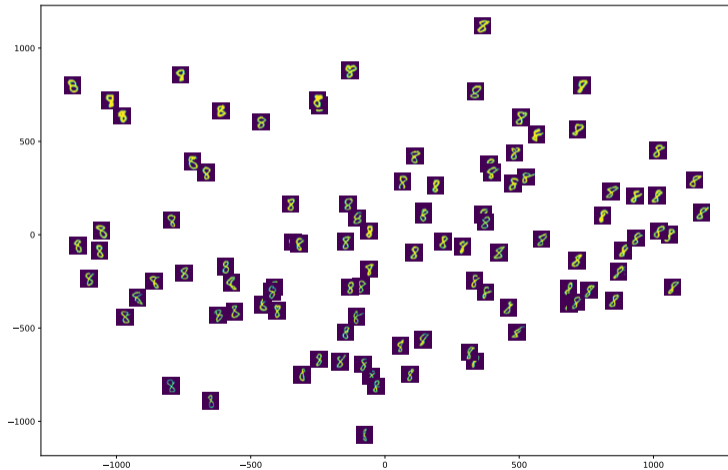
$$z_1 = \vec{x} \cdot \vec{u}^{(1)}$$

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

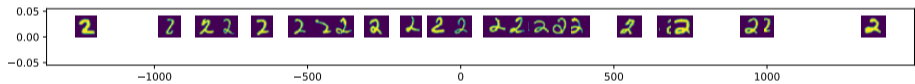
Example



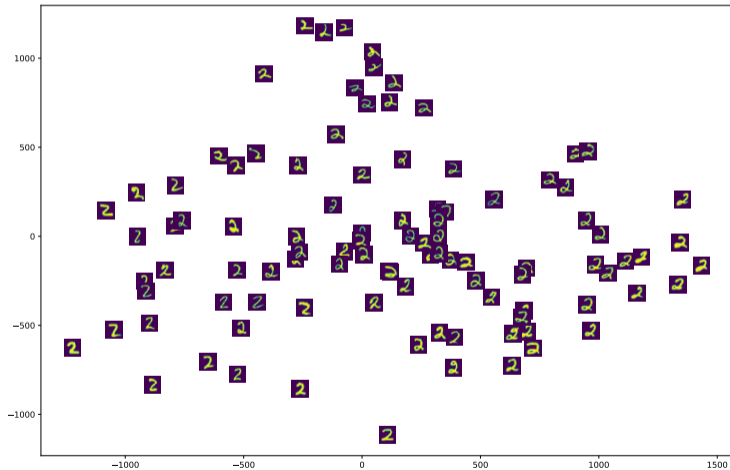
Example



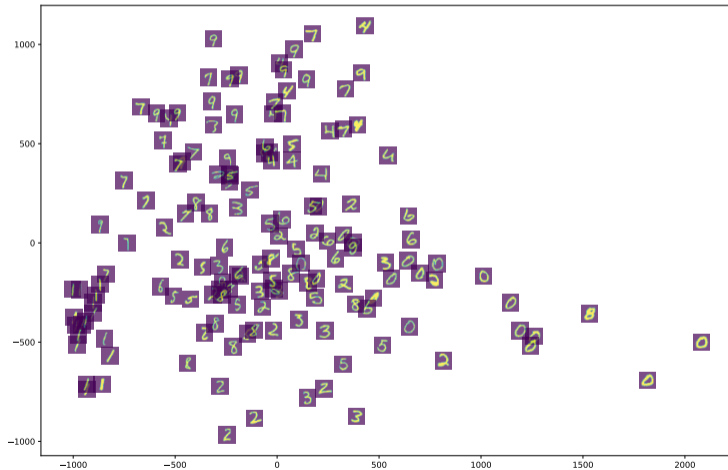
Example



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 190

Machine Learning: Representations

Lecture 9 | Part 3

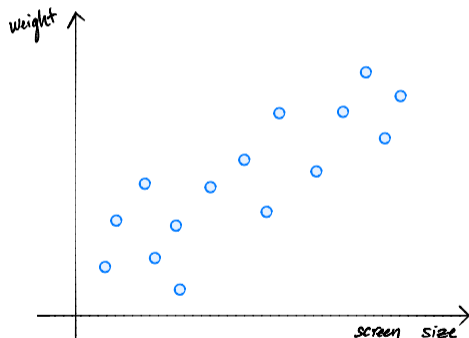
Reconstructions

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

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

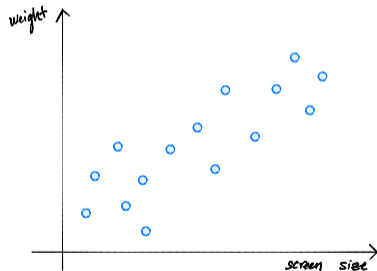
Reconstruction Error

- ▶ 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:

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



DSC 190

Machine Learning: Representations

Lecture 9 | Part 4

Interpreting PCA

Three Interpretations

- ▶ What is PCA doing?
- ▶ Three interpretations:
 1. Maximizing variance
 2. Finding the best reconstruction
 3. Decorrelation

Recall: Matrix Formulation

- ▶ Given data matrix X .
- ▶ Compute new data matrix $Z = XU$.
- ▶ PCA: choose U to be matrix of eigenvectors of C .
- ▶ For now: suppose U can be anything – but columns should be orthonormal
 - ▶ Orthonormal = “not redundant”

View #1: Maximizing Variance

- ▶ This was the view we used to derive PCA
- ▶ Define the **total variance** to be the sum of the variances of each column of Z .
- ▶ Claim: Choosing U to be top eigenvectors of C maximizes the total variance among all choices of orthonormal U .

Main Idea

PCA maximizes the total variance of the new data. I.e., chooses the most “interesting” new features which are not redundant.

View #2: Minimizing Reconstruction Error

- ▶ Recall: total reconstruction error

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

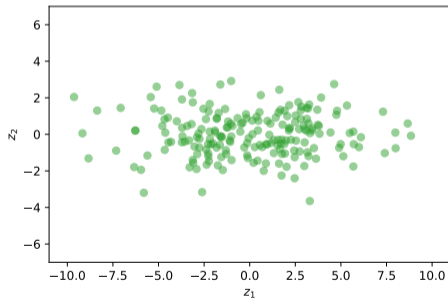
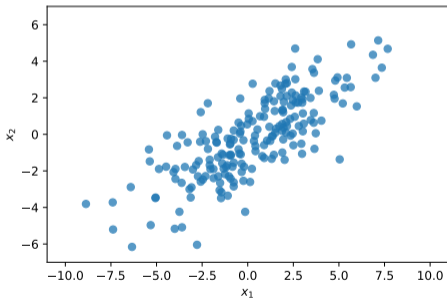
- ▶ Goal: minimize total reconstruction error.
- ▶ Claim: Choosing U to be top eigenvectors of C minimizes reconstruction error among all choices of orthonormal U

Main Idea

PCA minimizes the reconstruction error. It is the “best” projection of points onto a linear subspace of dimensionality k . When $k = d$, the reconstruction error is zero.

View #3: Decorrelation

- ▶ PCA has the effect of “decorrelating” the features.



Main Idea

PCA learns a new representation by rotating the data into a basis where the features are uncorrelated (not redundant). That is: the natural basis vectors are the principal directions (eigenvectors of the covariance matrix). PCA changes the basis to this natural basis.

DSC 190

Machine Learning: Representations

Lecture 9 | Part 5

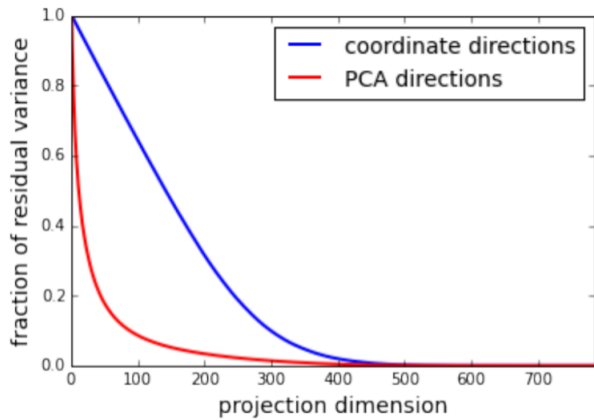
PCA in Practice

PCA in Practice

- ▶ PCA is often used in **preprocessing** before classifier is trained, etc.
- ▶ Must choose number of dimensions, k .
- ▶ One way: cross-validation.
- ▶ Another way: the elbow method.

Total Variance

- ▶ The **total variance** is the sum of the eigenvalues of the covariance matrix.
- ▶ Or, alternatively, sum of variances in each orthogonal basis direction.



Caution

- ▶ PCA's assumption: variance is interesting
- ▶ PCA is totally unsupervised
- ▶ The direction most meaningful for classification may not have large variance!

Demos