

Introduction to Machine Learning

Lecture 11 Unsupervised Learning I Dimensionality Reduction

Goker Erdogan
26 – 30 November 2018
Pontificia Universidad Javeriana

Types of learning problems

Supervised learning	Unsupervised learning	Reinforcement learning
<p>Learning from supervision (i.e., teacher)</p> <p>Given inputs and outputs, learn a function that maps input → output</p>	<p>Learning with no supervision</p> <p>Given only inputs, extract some pattern from the data.</p>	<p>Learning from rewards and punishments.</p> <p>Inspired by (operant) conditioning in psychology.</p> <p>The agent acts in an environment over some time.</p>
<p>Examples:</p> <ul style="list-style-type: none">- Recognizing faces- Predicting the sales for a product- Learning to rank search results	<p>Examples:</p> <ul style="list-style-type: none">- Clustering (market segmentation)- Dimensionality reduction (visualization)- Generating celebrity faces	<p>Examples:</p> <ul style="list-style-type: none">- Most robotics applications (e.g. learning to move around)- Playing games (e.g., AlphaGo)

Unsupervised learning

- Unsupervised learning is **difficult but important**
 - No clear objective
 - Hard to assess/validate results
 - Most of our learning is unsupervised

- **"Pure" Reinforcement Learning (cherry)**

- ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**

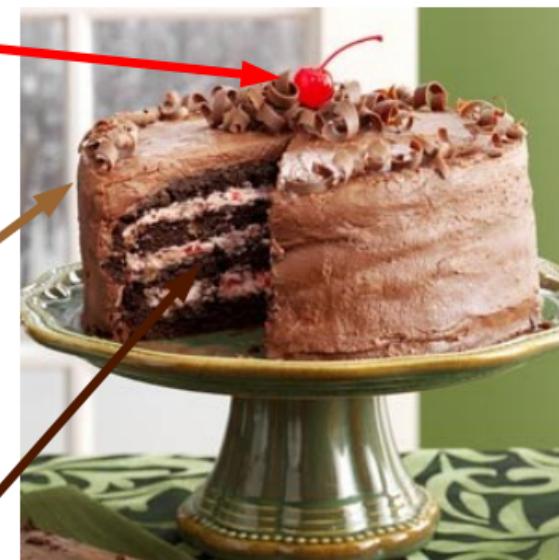
- **Supervised Learning (icing)**

- ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**

- **Unsupervised/Predictive Learning (cake)**

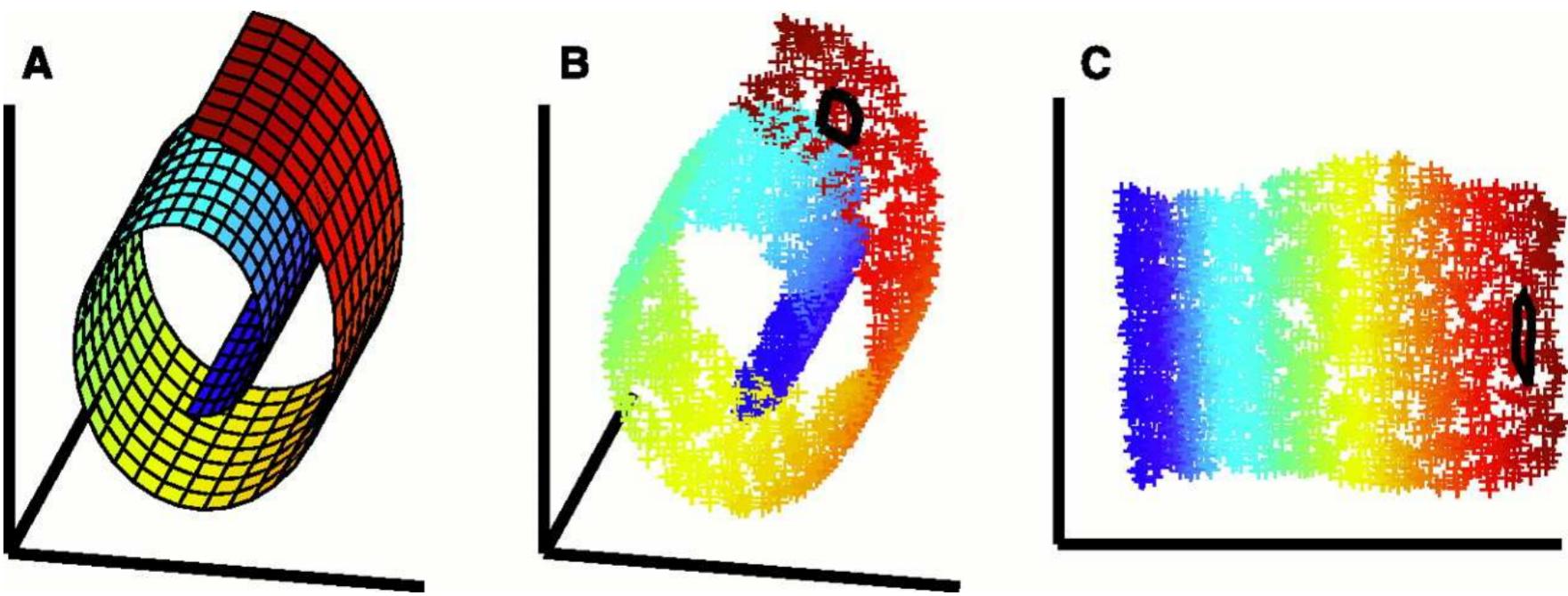
- ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ **Millions of bits per sample**

- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)



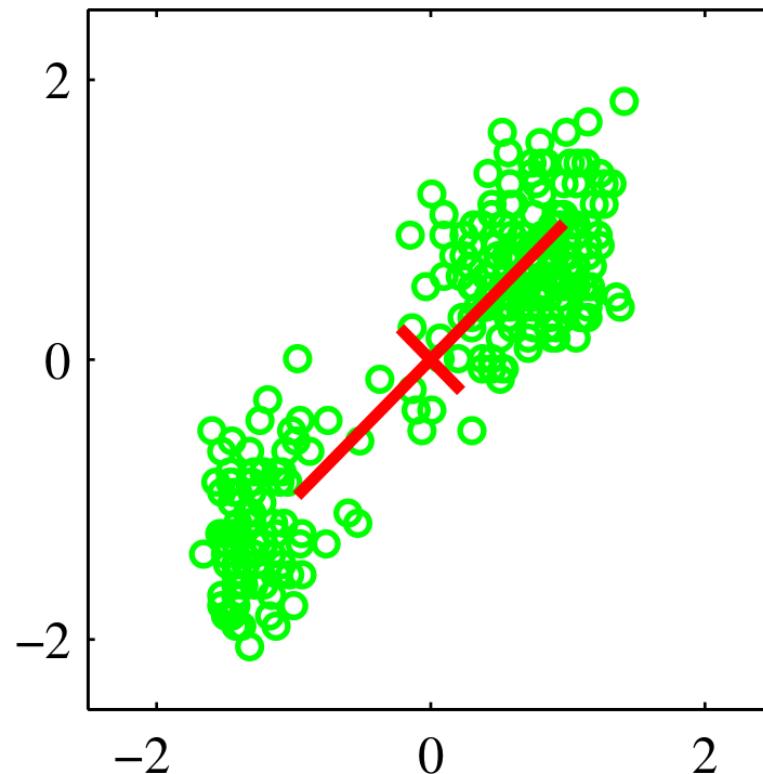
Dimensionality reduction

- Given a dataset of N samples with D features, X_{NxD}
 - Find a **low dimensional $M < D$ representation**
 - Capture as much of the information as possible
 - Need to define what we mean by information
- Why should this work?
 - Most data is **redundant** (e.g., images)
 - Correlated features
 - Intrinsic dimensionality
- Useful for
 - Visualization
 - Compression
 - Supervised learning



Principal components analysis (PCA)

- Popular technique for dimensionality reduction
 - Linear
 - Objective: Find direction(s) that maximize variance



PCA formulation

- Given N samples with D features, $X_{N \times D}$
 - Map each sample to a single number (1D)

$$z_n = \sum_u u_i X_{ni}$$
$$z = Xu$$

- Find direction (u) that maximizes variance

$$\max_u \sum_n z_n^2$$

$$\max_u (Xu)^T Xu$$

- Need to constrain u

$$\|u\|_2^2 = 1$$

$$u^T u = 1$$

Terminology:

u : loading vector

z : principal component

PCA formulation

$$\max_u (Xu)^T Xu$$

$$\text{s.t. } u^T u = 1$$

- Solution:

$$X^T X u = \lambda u \quad \lambda \in \mathbb{R}$$

- Known as an **eigenvector problem**
 - Note $X^T X$ is the **covariance matrix of X**

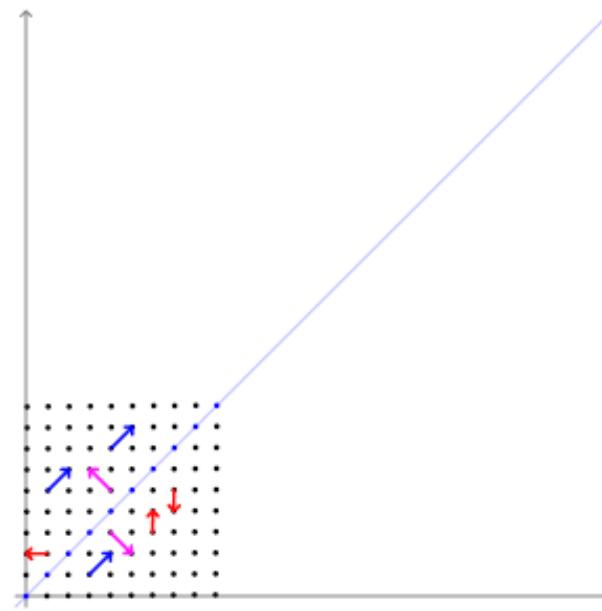
$$(X^T X)_{ij} = \mathbb{E}[X_{\cdot i} X_{\cdot j}]$$

- u is an **eigenvector** of $X^T X$
- λ is the **eigenvalue** associated with u
- Assumption: X is zero mean. $\frac{1}{N} \sum_n X_{ni} = 0, \forall i$

Eigenvectors

- Linear transformation A doesn't change direction of v

$$Av = \lambda v$$



- Eigendecomposition (for real symmetric matrices)

$$A = U\Lambda U^T$$

U : orthonormal

Λ : diagonal

Back to PCA

$$X^T X u = \lambda u$$

- Which eigenvector u ?

$$\begin{aligned} \text{Var}(z) &= (Xu)^T Xu \\ &= \lambda \end{aligned}$$

- Pick u with largest eigenvalue
- How do you pick the next principal components?
 - Find direction that maximizes variance
 - But uncorrelated with u
 - Pick u_2 orthogonal to u_1

$u_2 = \text{eigenvector with second largest eigenvalue}$
 - Similarly for u_3, u_4, \dots

Finding principal components

PCA

- Given input X_{NxD}
- Zero-mean and scale X (standardize)
- Reduce to $M < D$ dimensions (U_{DxM})

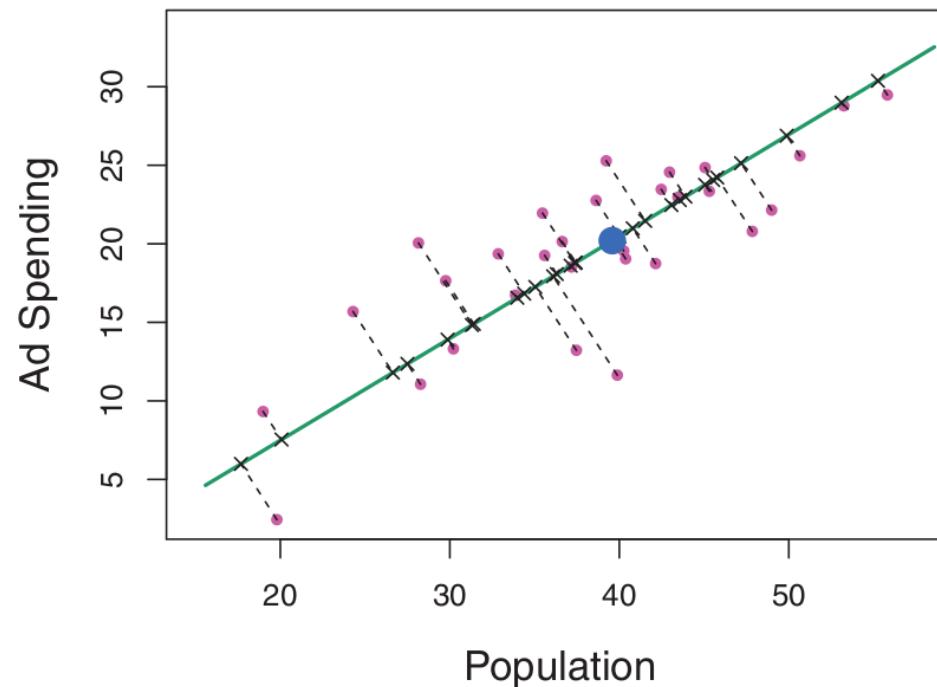
$$Z = XU$$

- Find eigenvectors of $X^T X$
- Pick the eigenvectors with largest M eigenvalues

- More efficient algorithms available
 - No need to calculate the full eigendecomposition
 - Only calculate M
 - If $D \gg N$, calculate using XX^T

An alternative perspective on PCA

- Maximizing variance = Minimizing reconstruction error
 - Find a low dimensional surface that is closest to samples
 - In terms of squared error
 - i.e., for 1D, find the line that minimizes error



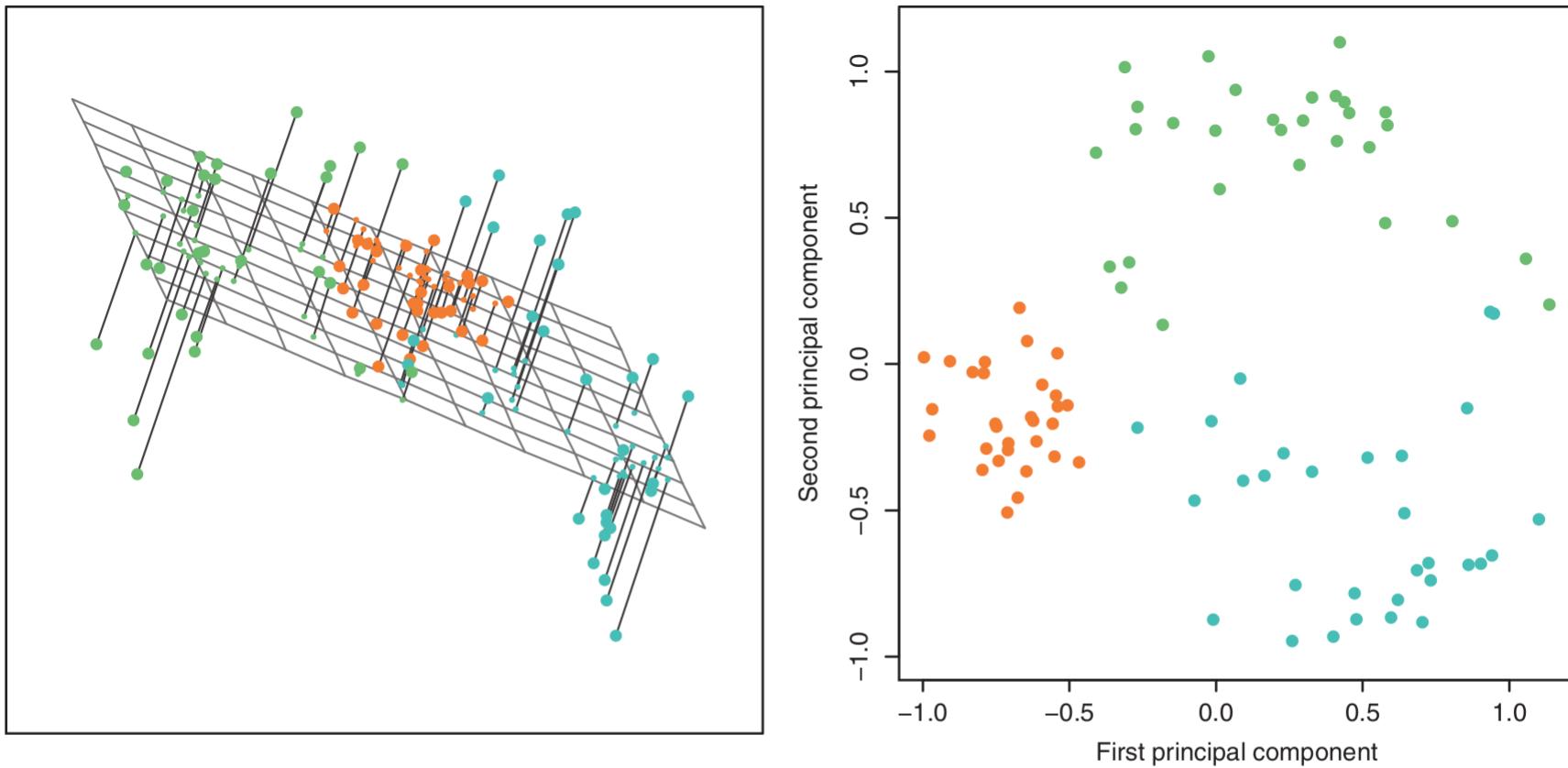


FIGURE 10.2. Ninety observations simulated in three dimensions. Left: the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane. Right: the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.



FIGURE 14.22. A sample of 130 handwritten 3's shows a variety of writing styles.

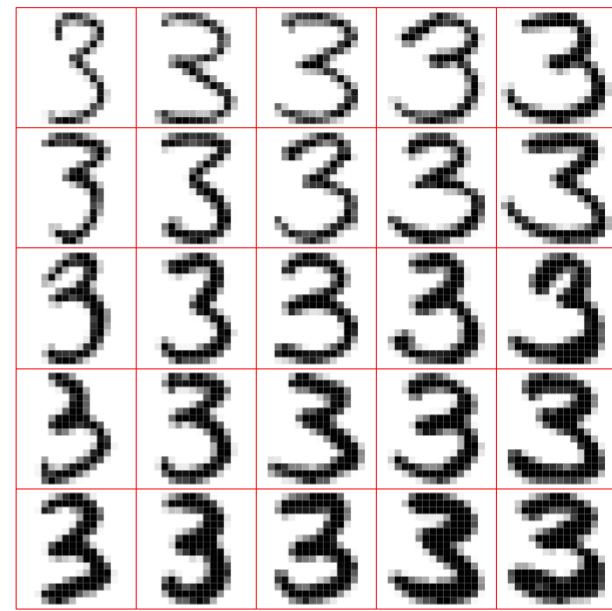
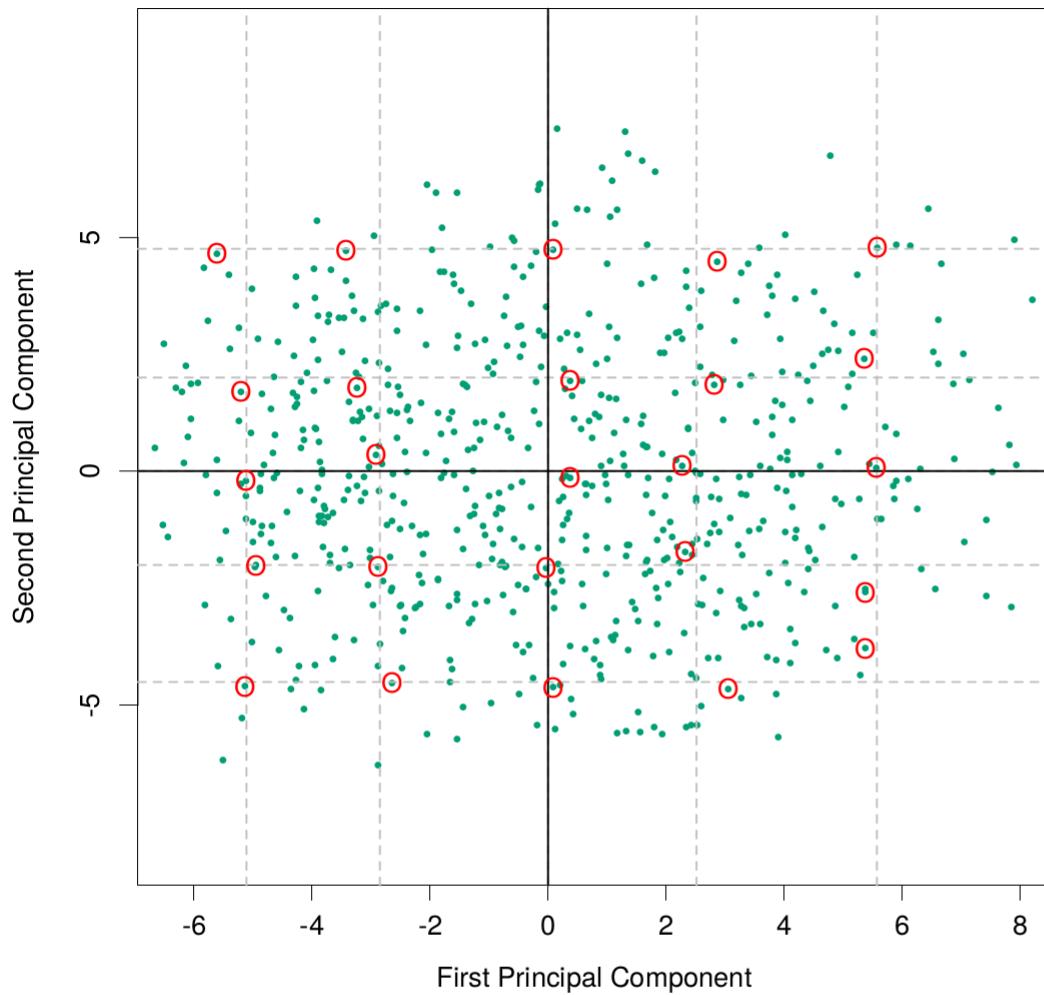


FIGURE 14.23. (Left panel:) the first two principal components of the handwritten threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.

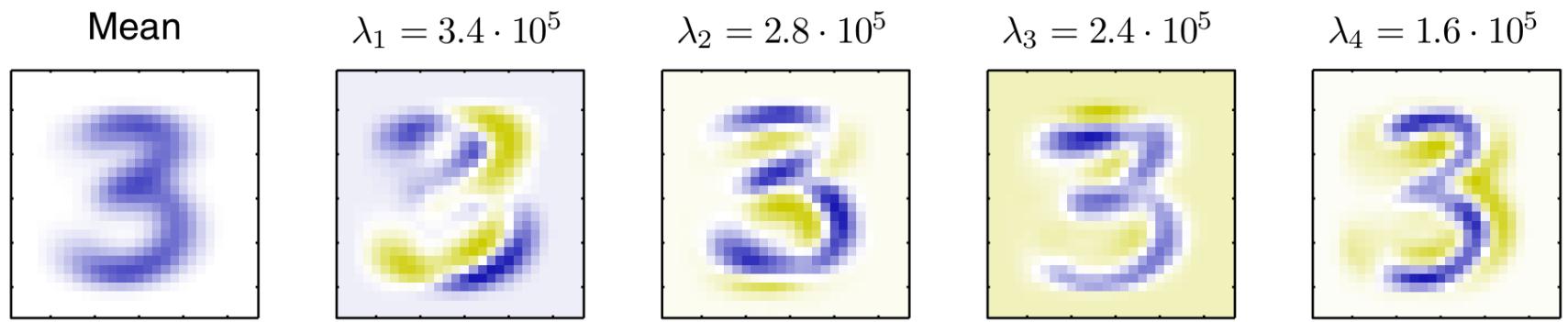


Figure 12.3 The mean vector \bar{x} along with the first four PCA eigenvectors u_1, \dots, u_4 for the off-line digits data set, together with the corresponding eigenvalues.

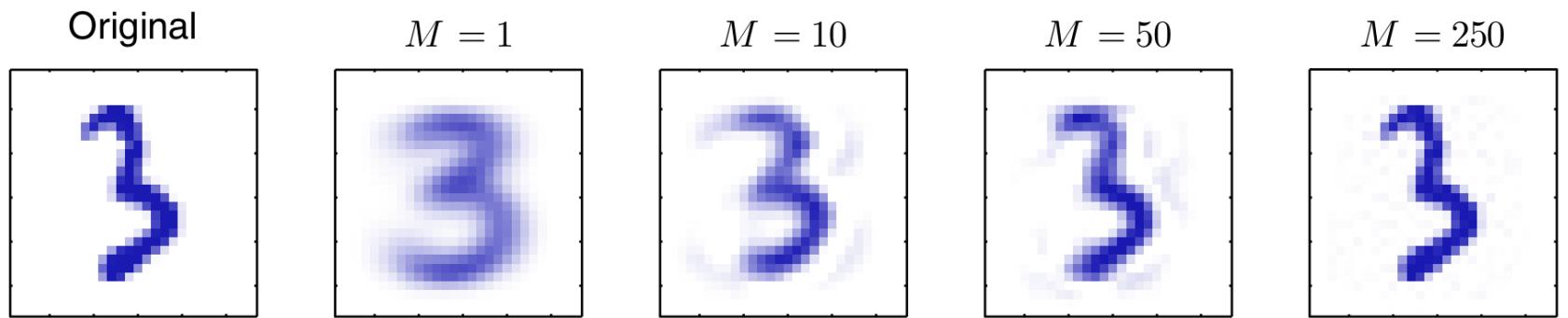


Figure 12.5 An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M . As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

Proportion of variance explained

- What should M be?
- Remember

$$\text{Var}(z_m) = \text{Var}((Xu_m^T)Xu_m) = \lambda_m$$

- Total variance:

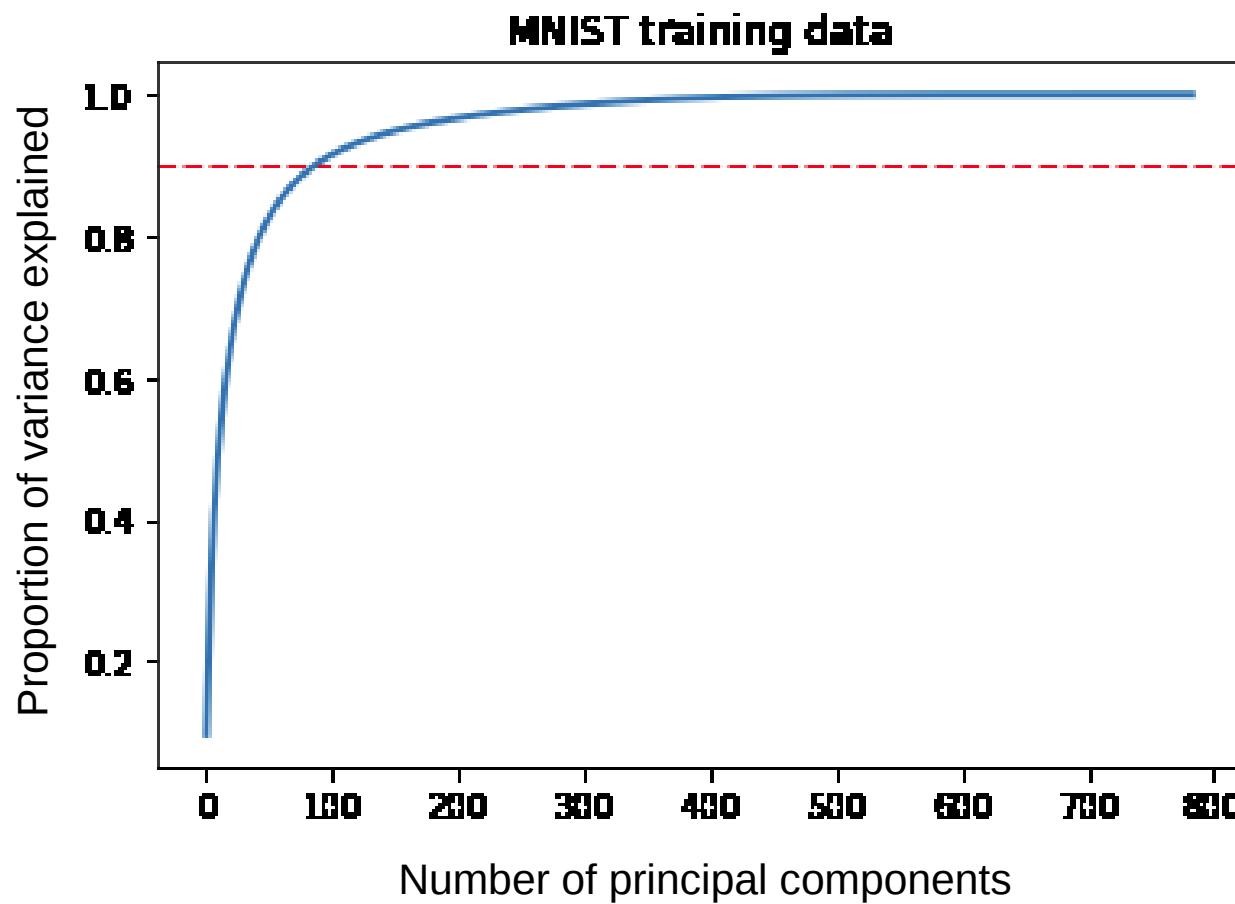
$$\sum_{m=1}^D \lambda_m$$

- Explained variance:

$$\sum_{m=1}^M \lambda_m$$

- Proportion of explained variance:

$$\frac{\sum_{m=1}^M \lambda_m}{\sum_{m=1}^D \lambda_m}$$



Whitening (sphering)

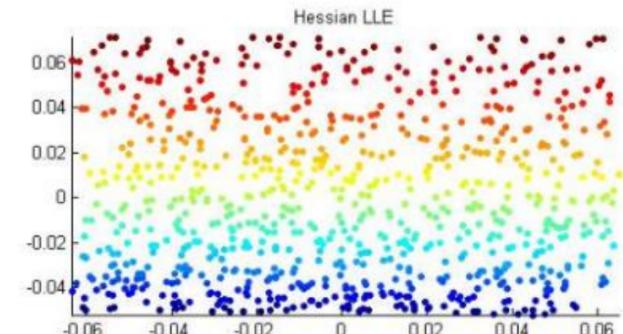
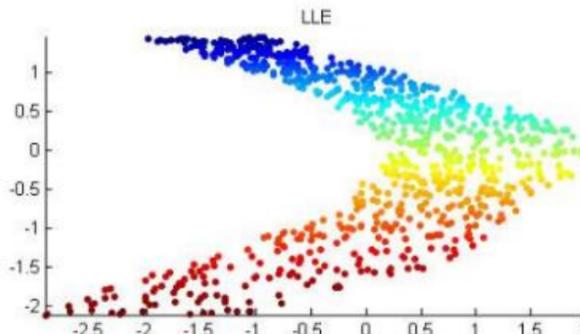
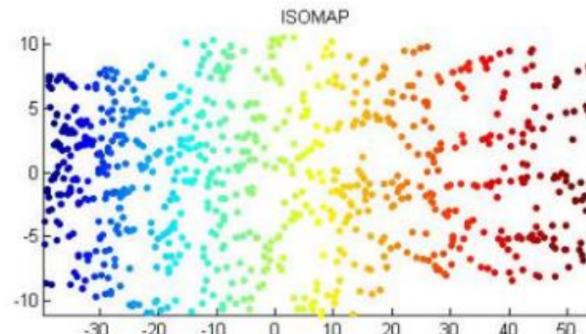
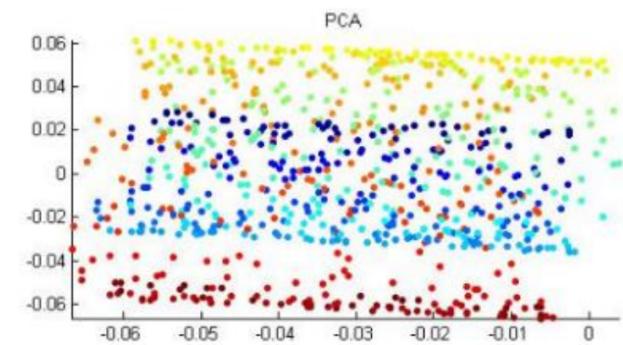
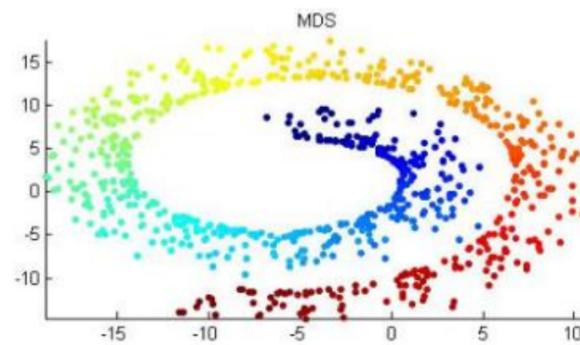
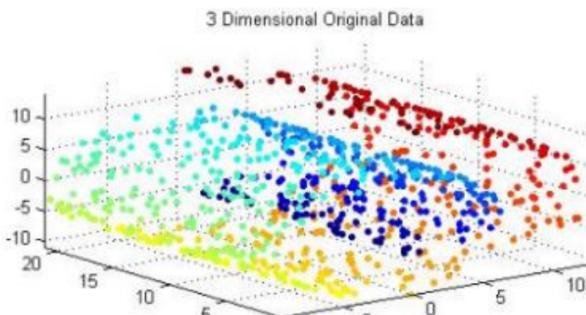
- Standardize inputs
 - Zero-mean, unit variance
 - But **features can be correlated**
- Whitening removes correlations
 - De-correlates input data

$$\tilde{X} = XU\Lambda^{-1/2}$$

- Uncorrelated features
- $$\text{Cov}[\tilde{X}] = \mathbf{I}$$
- Feed \tilde{X} instead of X into supervised technique

Beyond PCA

- Many non-linear dimensionality reduction techniques
 - Kernel PCA
 - ISOMAP
 - t-SNE
 - Locally linear embedding
 - Spectral clustering



Summary

- Dimensionality reduction
- Principal components analysis
 - Formulation: maximize variance
 - Minimize reconstruction error
 - Proportion of variance explained
- Beyond PCA
- Exercises
 - Show that the direction that maximizes variance is the eigenvector of covariance matrix with largest eigenvalue
 - Do lab 10.4 in ISLR

References

- [1] James, Witten, Hastie, and Tibshirani. An Introduction to Statistical Learning with Applications in R. Chapter 10.
- [2] Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning. Chapter 14.
- [3] Bishop, C. Pattern Recognition and Machine Learning. Chapter 12
- [4] LeCun, Y.
- [5] <http://statweb.stanford.edu/~tibs/sta306bfiles/isomap.pdf>
- [6] https://www.researchgate.net/figure/PCA-of-MNIST-FIG-4-PCA-of-not-MNIST_fig2_320517142