Principal Component Analysis

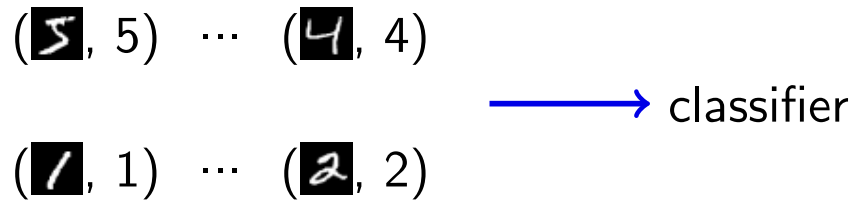
Nan Ye

School of Mathematics and Physics
The University of Queensland

Recall

Supervised Learning

- Fit a model relating x and y given a dataset $(x_1, y_1), \ldots, (x_n, y_n)$
- In a classification problem, the output is discrete.



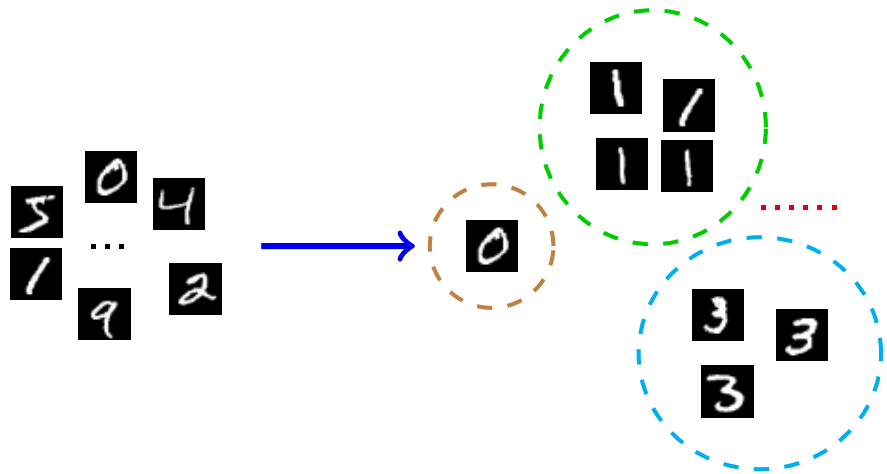
- In a regression problem, the output is real-valued.

Terminology

- *x: input, independent variables, covariate vector, observation, predictors, explanatory variables, features.*
- *y: output, dependent variable, response.*

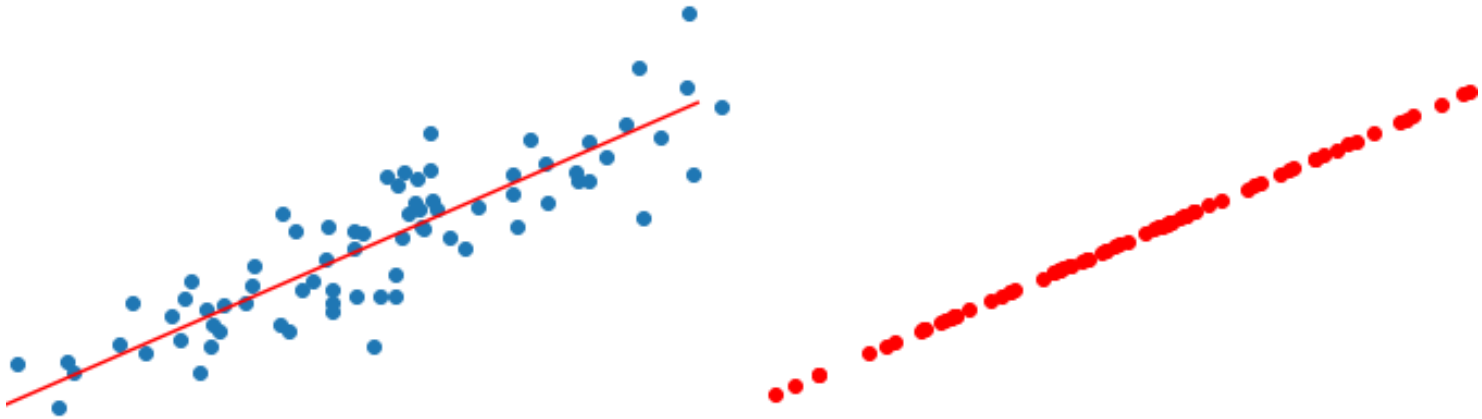
Unsupervised Learning

- Only the inputs are given, but not the outputs
- Unsupervised learning methods are used for various purposes, e.g.
  - Clustering: divide data points into groups



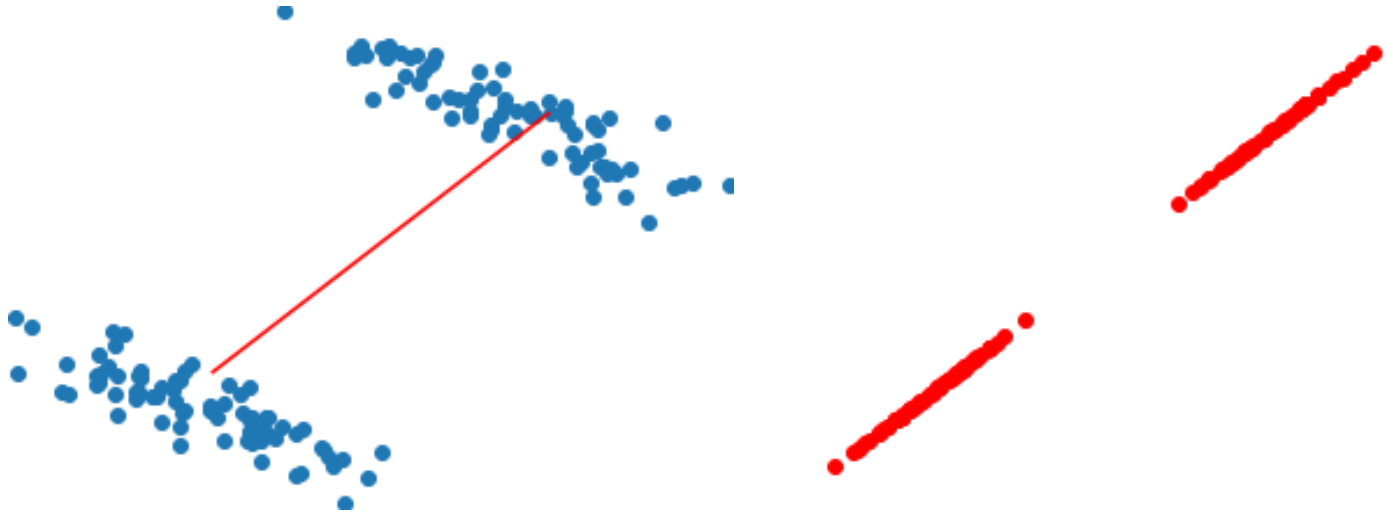
  - Density estimation: estimate a distribution given a sample
  - Dimension reduction: find a low-dimensional representation of data

Directions of Dominant Variations



(a) Original (b) Projection

- Difference between data points may be coming mainly from a few directions.
- In the example above, the data points mainly fluctuate along the red line, and we can consider their projections to the red line as a good approximation.
- This can help us filter out noise.



(a) Original (b) Projection

- In the example above, the data points are grouped into two distinct clusters, but overall, they show maximum variation along the red line.
- Projecting the data points on the red line is lossy, but still preserves the cluster information.

- In the above two examples, we only consider a single direction of maximum variation.
- In general, we can find additional directions of maximum variation in an incremental way, by choosing a new direction that is orthogonal to existing directions and maximizes the variance of the projections.

Principal Components Analysis (PCA)

- PCA takes in multi-dimensional data and finds orthogonal directions in which the data has the largest variance.
- These directions, called the principal components, form a lower-dimensional subspace.
- We can represent the original data points by their projections onto the principal directions.
- We lose all information about where the datapoint is located in the remaining orthogonal directions.

Principal components as eigvenvectors

- Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbf{R}^d$, if $\mathbf{v}$ is a unit vector such that the variance of their projections on $\mathbf{v}$ is maximised, then $\mathbf{v}$ satisfies

$$\Sigma \mathbf{v} = \lambda \mathbf{v},$$

  where $\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$ is the empirical covariance matrix.

- In addition, λ is the variance of the projections.
- In general, the top k eigenvectors of Σ are known as the top k principal components.
  - *top k = have largest k eigenvalues*

Why?

- The variance of the projections on $\mathbf{v}$ is given by

$$\frac{1}{n}\sum_{i=1}^{n}[\mathbf{v}^\top(\mathbf{x}_i - \bar{\mathbf{x}})]^2 = \mathbf{v}^\top\Sigma\mathbf{v}. \quad (\textit{verify})$$

- Thus we want to maximize $\mathbf{v}^\top\Sigma\mathbf{v}$ subject to $||\mathbf{v}||_2^2 = 1$.
- The Lagrangian and its gradient with respect to $\mathbf{v}$ are

$$L(\mathbf{v}, \lambda) = \mathbf{v}^\top\Sigma\mathbf{v} - \lambda(||\mathbf{v}||_2^2 - 1),$$
$$\nabla_{\mathbf{v}} L = 2\Sigma\mathbf{v} - 2\lambda\mathbf{v}. \quad (\textit{verify})$$

From the 2nd equation, we have $\Sigma\mathbf{v} = \lambda\mathbf{v}$. In addition, this implies that the variance along $\mathbf{v}$ is $\mathbf{v}^\top\Sigma\mathbf{v} = \mathbf{v}^\top(\lambda\mathbf{v}) = \lambda$.

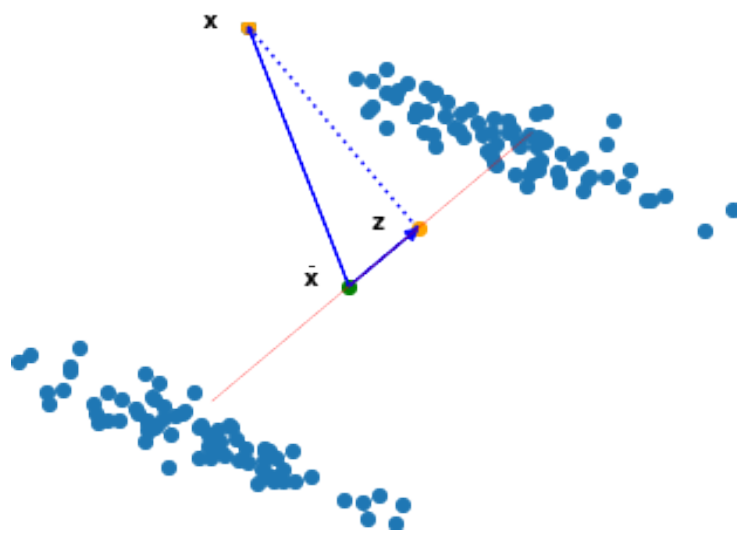
Properties

- The principal components are orthogonal.
- The projections of the data on two different principal components are linearly uncorrelated.

  this does not imply that the projections are uncorrelated — one may be the nonlinear function of another!

Dimension reduction using PCA

- Input: $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbf{R}^d$, dimension k.
- Output: k-dimensional representation $\mathbf{z}_i$ of $\mathbf{x}_i$, for each i.
- Procedure
  - $\mathbf{v}_1, \ldots, \mathbf{v}_k \leftarrow$ top k principal components
  - $\mathbf{z}_i \leftarrow ((\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{v}_1, \ldots, (\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{v}_k)$, for each i.



Reconstruction

- Input: principal components $\mathbf{v}_1, \ldots, \mathbf{v}_k$, reduced data representations $\mathbf{z}_1, \ldots, \mathbf{z}_n$.
- Output: $\tilde{\mathbf{x}}_i \approx \mathbf{x}_i$ for each i.
- Procedure: $\tilde{\mathbf{x}}_i \leftarrow \bar{\mathbf{x}} + \sum_{j=1}^{k} z_{ij}\mathbf{v}_j$, for each i.

PCA vs SVD (optional)

- Principal components are often found by performing SVD on the centered design matrix.
- Specifically, the empirical covariance matrix Σ can be written as

$$\Sigma = \frac{1}{n}\bar{\mathbf{X}}^\top \bar{\mathbf{X}},$$

where $\bar{\mathbf{X}} \in \mathbf{R}^{n \times d}$ is the centered design matrix, that is, the i-th row of $\bar{\mathbf{X}}$ is $\mathbf{x}_i - \bar{\mathbf{x}}$.

- Given the SVD $\bar{\mathbf{X}} = U\Lambda V^\top$, where $U \in \mathbf{R}^{n \times n}$ and $V \in \mathbf{R}^{d \times d}$ are orthogonal matrices, and $\Lambda \in \mathbf{R}^{n \times d}$ is a diagonal matrix, we have

$$\Sigma = \frac{1}{n} V\Lambda^\top U^\top U\Lambda V^\top = VDV^\top \qquad \Rightarrow \qquad \Sigma V = VD,$$

where $D = \frac{1}{n}\Sigma^\top \Sigma \in \mathbf{R}^{d \times d}$ is a diagonal matrix with the diagonal entries being $1/n$ of the squares of the singular values.

- Hence the columns of V are exactly the eigenvectors of Σ (the principal components), and the diagonal entries of D are the corresponding eigenvalues (variances of projections).

Your turn

Given three feature vectors $(1, 2)$, $(2, 3)$ and $(5, 6)$.

- Draw the scatter plot for them, and show the directions of the two principal components.
- If we only use the top principal component for reconstruction, can we perfectly reconstruct the original data?

Image Compression

- Consider an image of size $h \times w$.
- We can view the rows of the image as the data points, and compute their top k principal components.
- Now we represent each row using its coordinates in the subspace with the k principal components as the basis.
- While the original image has size $h \times w$, the compressed representation has size $(k+1)w + kh$.

(a) Original **(b)** $k = 10$ **(c)** $k = 50$ **(d)** $k = 100$

- The original image has size 512x512x3 (think of this as 512x1536).
- With $k = 50$, we can reduce the image size by a factor or 512x1536/(51x1536 + 50x512)=7.5665 (i.e., a compression ratio of 7.5665).

Eigenface

- Consider a database of face images.
- We can view each image as a vector, and find the top principal components for them.
- With these principal components, we can reduce the dimension of the images, and also reconstruct them from the lower dimensional representation.
- Each principal component can be viewed as an image. These are called the eigenfaces.

(a) Original images

(b) PCA reconstruction (32 components)

The reconstruction is lossy (the two reconstructed images in red boxes are hardly distinguishable).

- Top eigenfaces are blurry and have no distinct features.

(a) Original images

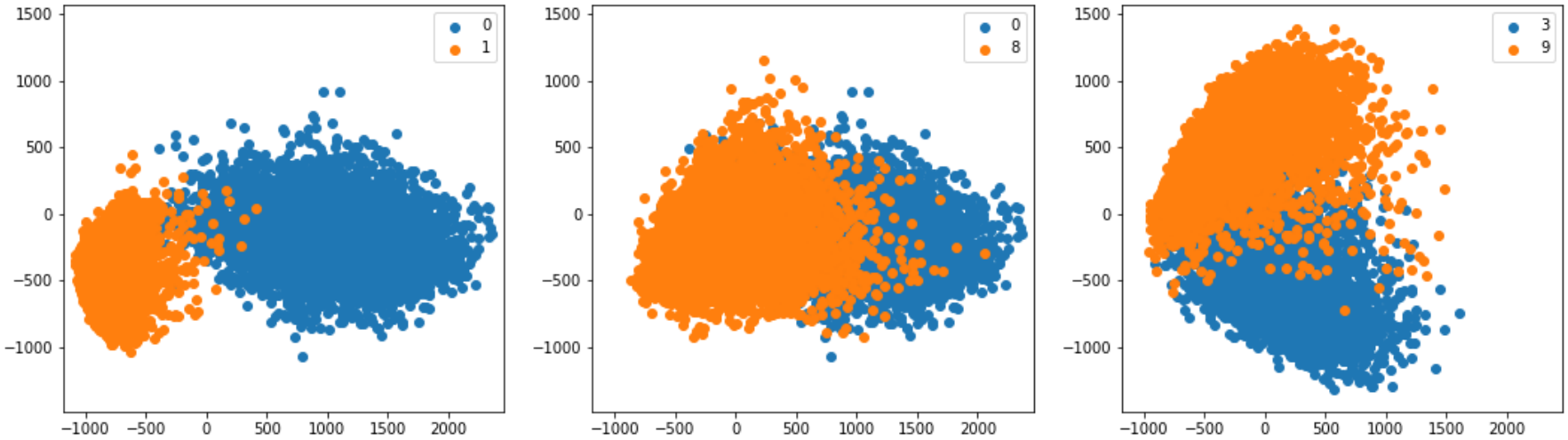
(b) PCA reconstruction (72 components)

With more components, PCA starts to pick up very fine features (compare the two images in red boxes again).

PCA + Supervised Learning

- For supervised learning, we can first apply PCA to reduce the dimension of the data, and then perform supervised learning on the lower dimensional data.
  - e.g. train a classifier for face recognition using the lower dimensional representation
- This can be used as a way to filter out irregularities in the dataset, thus helps in preventing the learning algorithm to fit to irregularities.
- We can thus view PCA as an implicit regularization technique.

Visualizing Classifiers



- The plots show how some classes in the MNIST training data look like when projected to the subspace spanned by the top 2 eigenvectors.
- While we lose quite a bit of information, the embedding still gives us some idea of how well the classes are separated.
- We can use such plots to visualize any classifier as well.

More on Dimension Reduction

- Two types of methods
  - Feature selection: find a subset of most important variables.
    - Lasso, LARS, forward/backward selection,...
  - Feature extraction (or feature projection): embed/project the data to a lower dimensional space.
    - PCA, kernel PCA, Isomap, multidimensional scaling, t-SNE, LDA, autoencoder, ...
- We will cover autoencoder in this course (also useful for image compression).

What You Need to Know...

- Principal components
  - Intuition: directions along which the data varies most
  - Eigenvectors of the empirical covariance matrix
- Reduced representation and reconstruction
- Applications
  - Image compression, eigenface
  - PCA + supervised learning
  - Visualizing classifiers