Principle component analysis

Ngô Minh Nhựt

2024

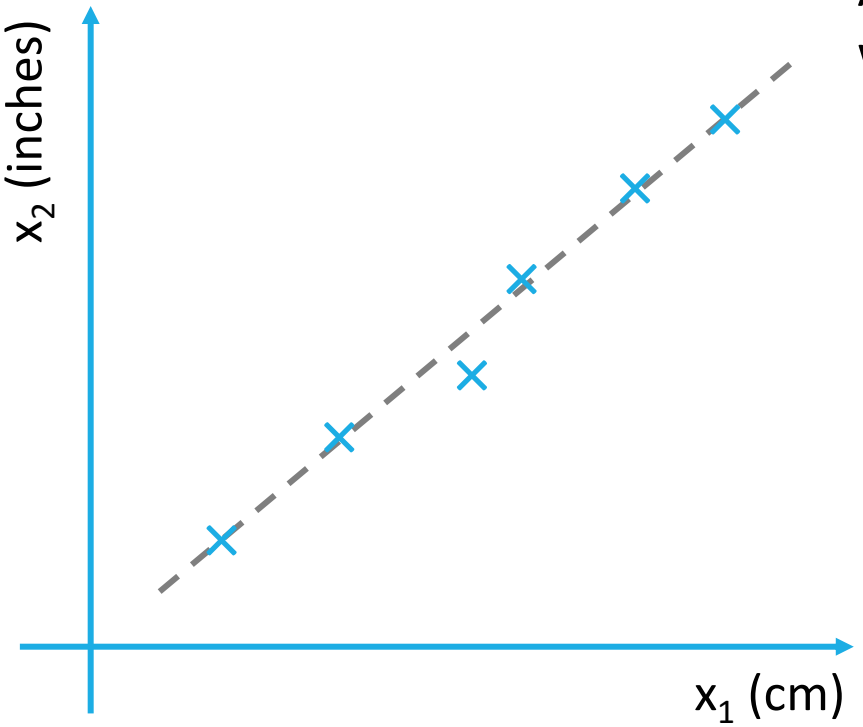
Outline

- Motivation
- Problem formulation
- PCA algorithm
- Applying PCA

Dimensionality reduction

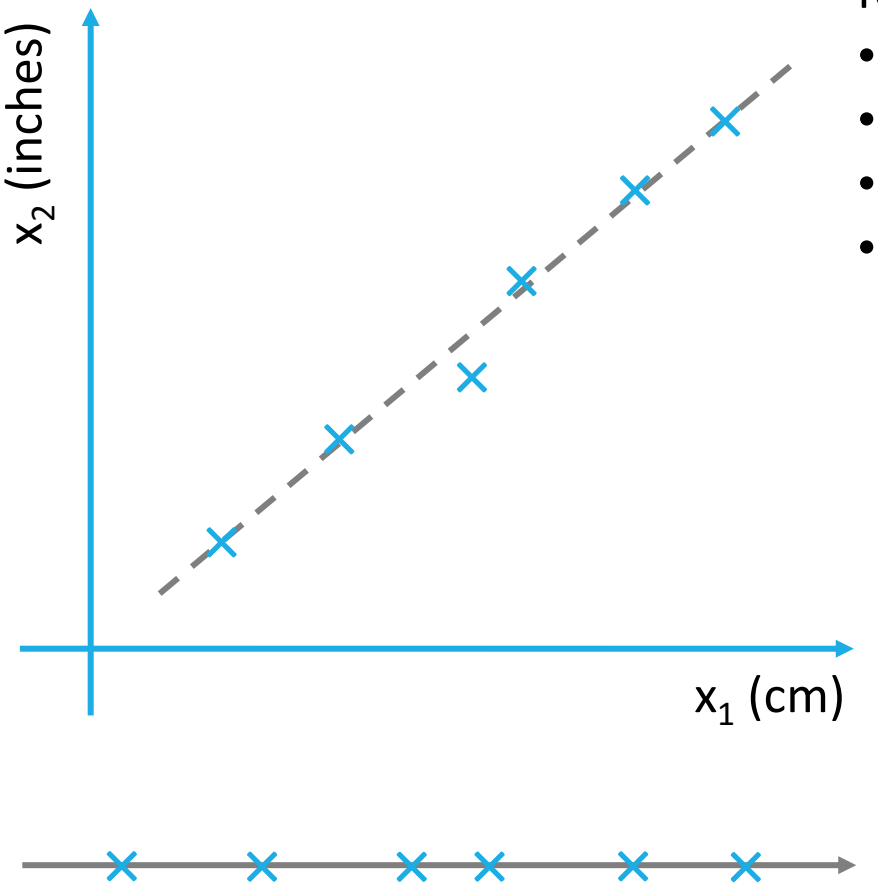
- Dimensionality reduction: map data to a lower dimensionality space
- Motivation
  - Data compression
  - Data visualization
- PCA is used for dimensionality reduction

Data compression



As a strong correlation exists,
we can reduce data from 2D to 1D

Data compression

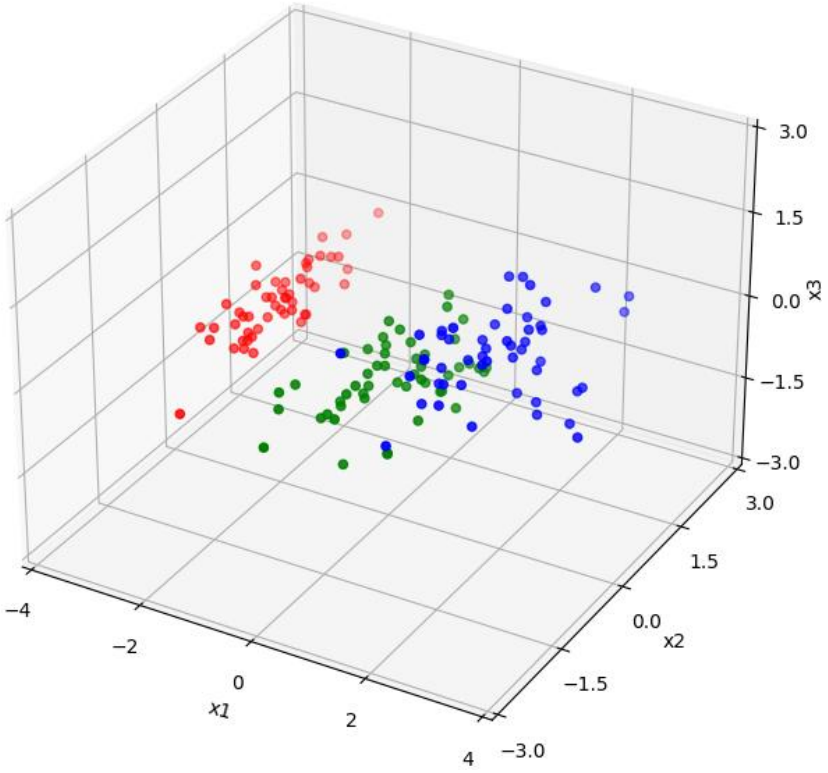


Reduce data from 2D to 1D

- $x^{(1)} \in \mathbb{R}^2 \rightarrow z^{(1)} \in \mathbb{R}$
- $x^{(2)} \in \mathbb{R}^2 \rightarrow z^{(2)} \in \mathbb{R}$
- …
- $x^{(m)} \in \mathbb{R}^2 \rightarrow z^{(m)} \in \mathbb{R}$

Project onto
1-dimension

Data compression



Project onto
2-dimension

Reduce data from 3D to 2D

- $x^{(1)} \in \mathbb{R}^3 \rightarrow z^{(1)} \in \mathbb{R}^2$
- $x^{(2)} \in \mathbb{R}^3 \rightarrow z^{(2)} \in \mathbb{R}^2$
- …
- $x^{(m)} \in \mathbb{R}^3 \rightarrow z^{(m)} \in \mathbb{R}^2$

Data visualization

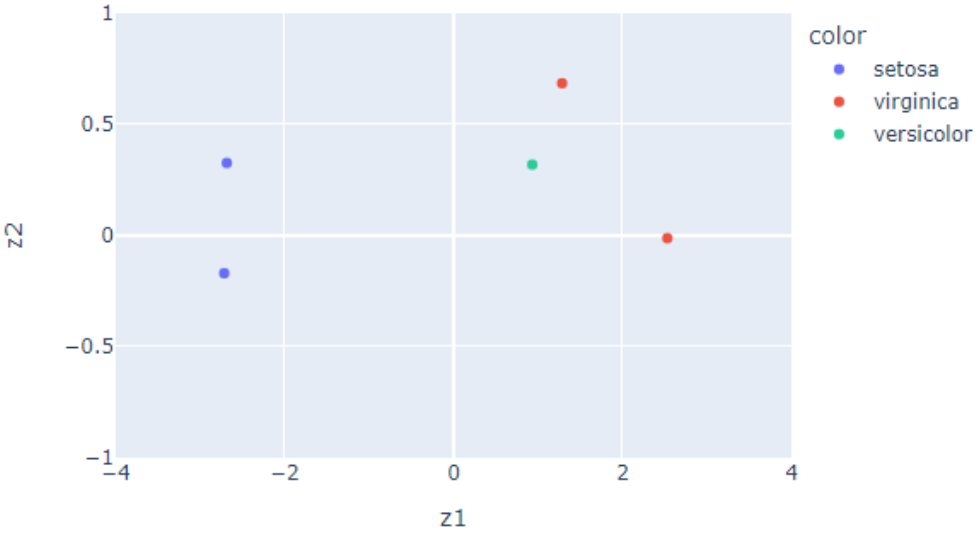
- Iris dataset of flowers of three species (classes) with four features

Sample	Sepal width	Sepal length	Petal width	Petal length	Class
0	3.5	5.1	0.2	1.4	setosa
1	3.0	4.9	0.2	1.4	setosa
2	3.3	6.3	2.5	6.0	virginica
3	3.2	7.0	1.4	4.7	virginica
4	3.2	6.4	1.5	4.5	versicolor
…					

Data visualization

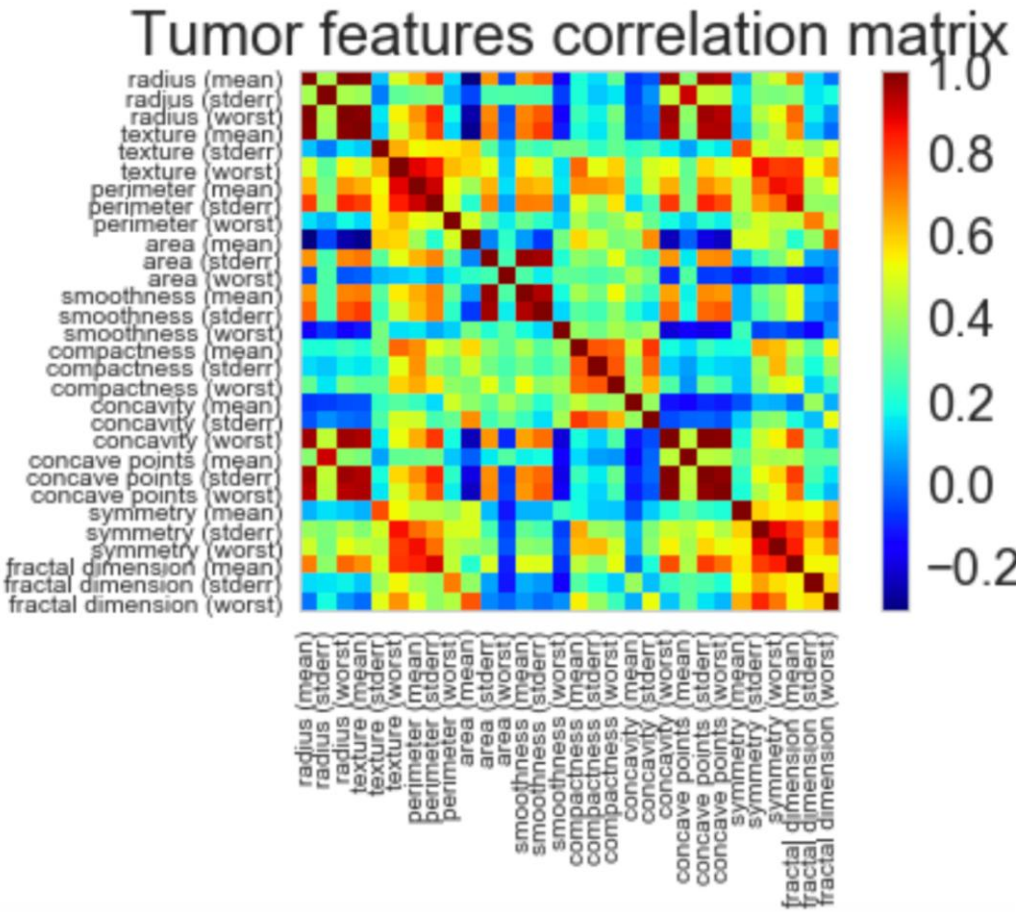
- When dimensionality of samples is reduced

Sample	z1	z2	Class
0	-2.68	0.33	setosa
1	-2.72	-0.17	setosa
2	2.5	-0.01	virginica
3	1.28	0.69	virginica
4	0.93	0.32	ver
…			



PCA – Problem formulation

- Predict if a breast tumor is malign or benign using data about tumor cell features



- The data includes 30 different cell features.
- There are many features that are highly correlated with each other.

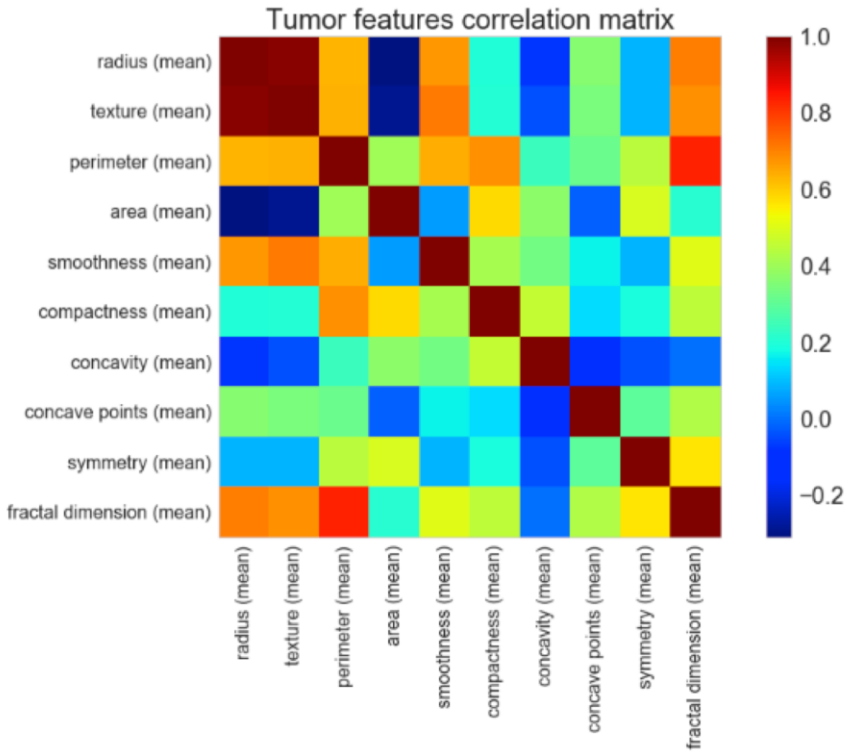
Reduce the feature space

PCA – Problem formulation

Reduce the feature space

- Approach 1: remove some of the features, e.g., keeping one the features regarding the mean

  - Pros: simple and maintain interpretation of the features
  - Cons: lose information from the features dropped



PCA – Problem formulation

Reduce the feature space

- Approach 2: get a new dataset, resulting from a linear combination of the original dataset

  - Pros: less features containing information of all features
  - Cons: new features no longer have meaningful interpretation (characteristic of tumor cell)

$$A = \begin{bmatrix} \vdots & \vdots & \vdots \\ F_1 & \dots & F_{30} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$A^* = \begin{bmatrix} \vdots & \vdots & \vdots \\ F_1^* & F_2^* & F_3^* \\ \vdots & \vdots & \vdots \end{bmatrix}$$

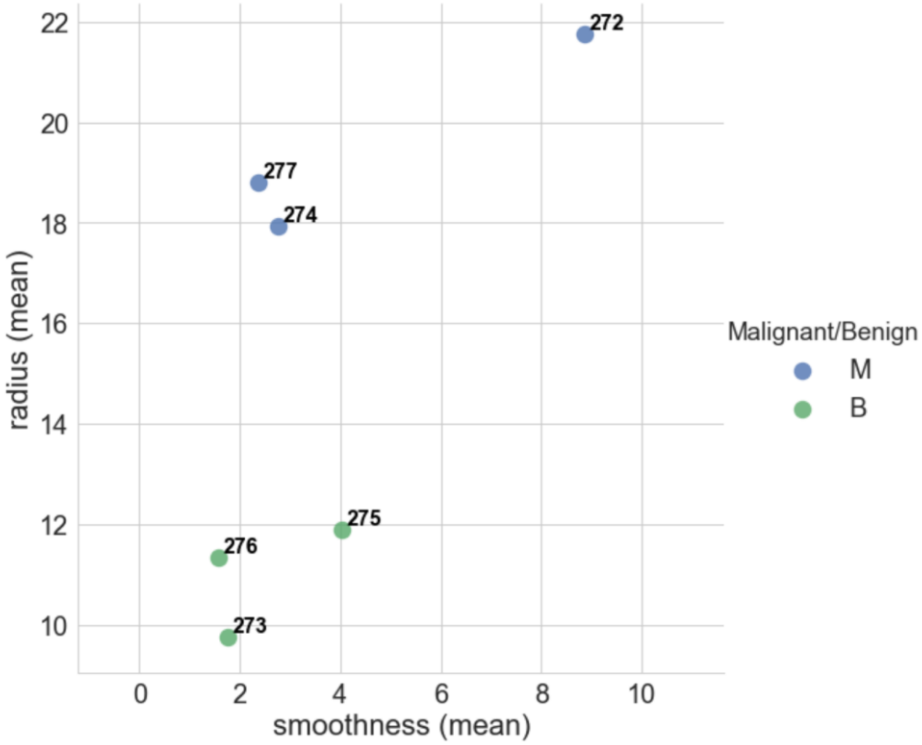
$$F_1^* = \sum_{i=1}^{n} a_i F_i$$

PCA – Problem formulation

- PCA will combine the features in a specific way, creating *new features*
- PCA allows dropping *least important* new features while still retaining most valuable parts of the original features
- As an added benefit, each of the new features after PCA are all independent each other (important for linear models)

PCA – Problem formulation

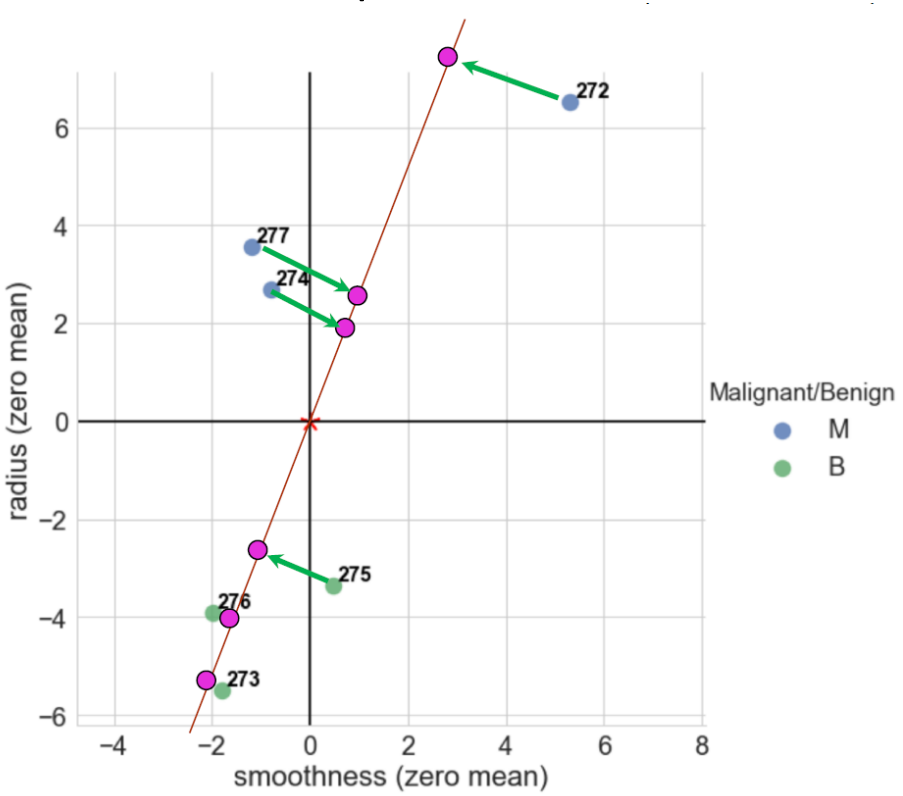
- Predict if a breast tumor is malign or benign using data about tumor cell features
- Start with six patients and two features: smoothness and radius



	smoothness (mean)	radius (mean)
272	8.867	21.750
273	1.750	9.742
274	2.765	17.930
275	4.021	11.890
276	1.565	11.330
277	2.363	18.810

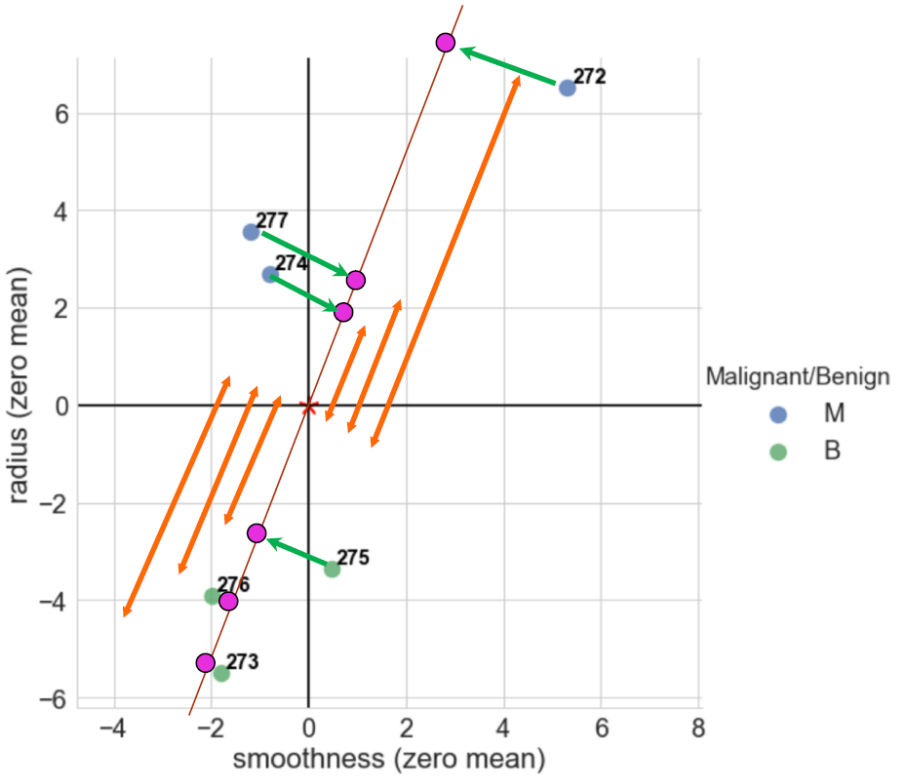
PCA – Problem formulation

- We want to find a line that fits the dataset
  - To quantify how good the fit is, PCA projects the data onto the line
  - The best fit minimizes the distances from the points to the line (indicated in green below)



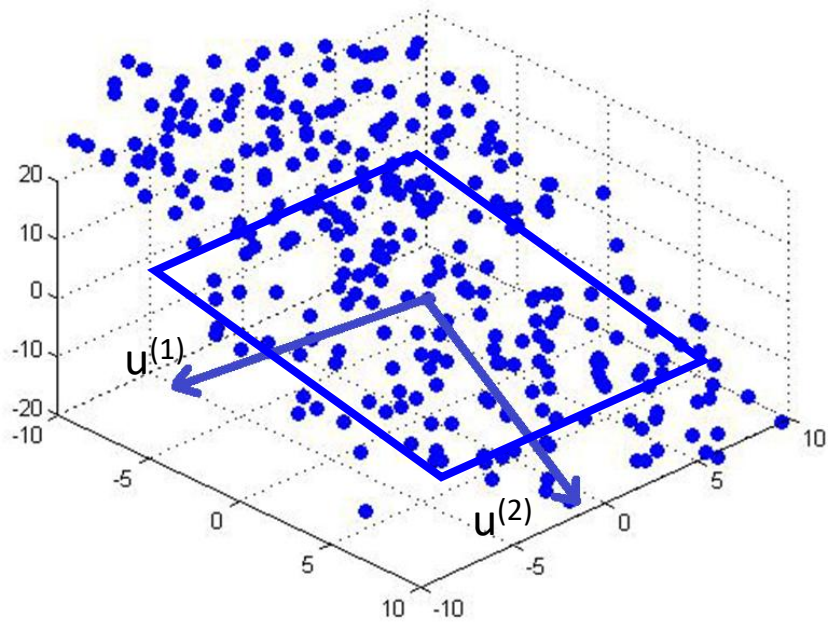
PCA – Problem formulation

- We want to find a line that fits the dataset
  - To quantify how good the fit is, PCA projects the data onto the line
  - Or maximizes the distances (indicated in orange) from the projected points to the origin

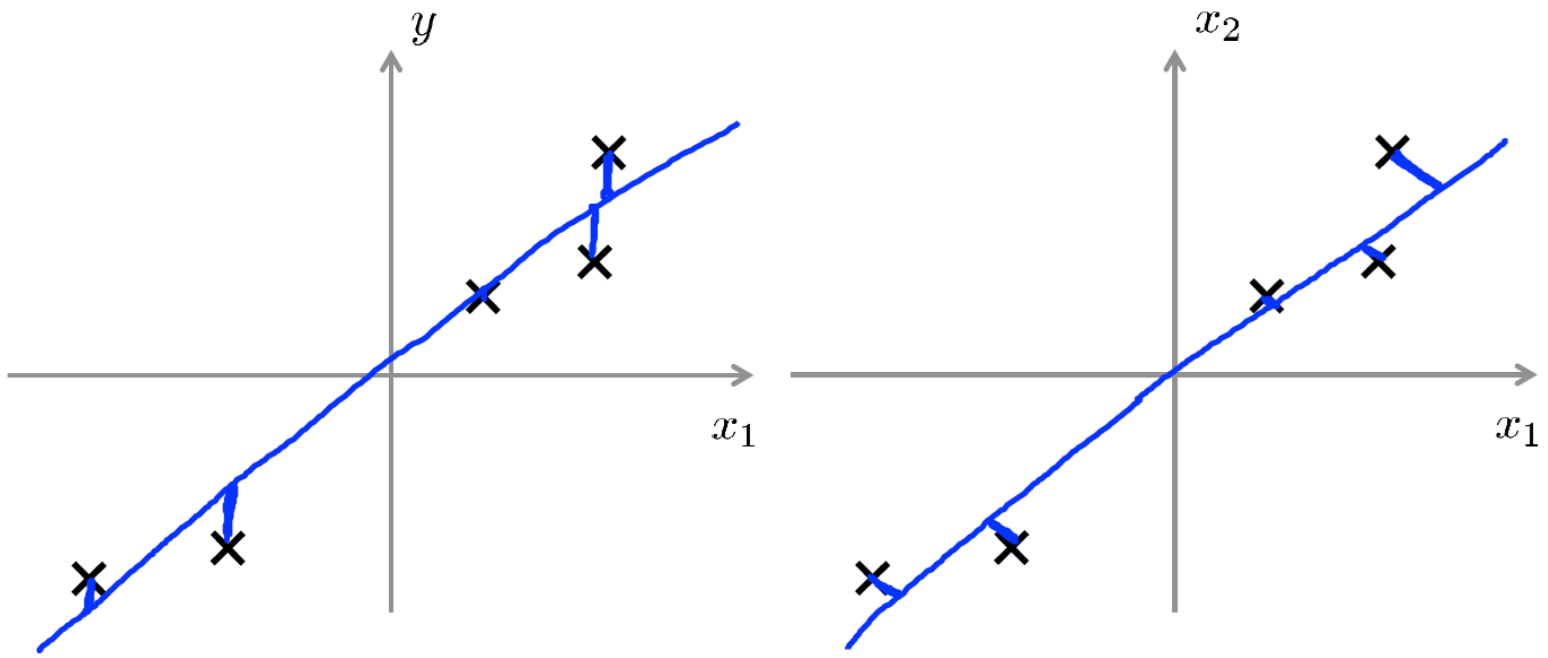


PCA – Problem formulation

- To reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^2$) onto which to project the data so as to minimize the projection error.
- To reduce from n-dimension to k-dimension: Find k direction $u^{(1)}$, $u^{(2)}$, …, $u^{(k)}$ onto which to project the data so as to minimize the projection error.



PCA is not linear regression



Which one is PCA and which one is linear regression?

PCA - Algorithm

Data preprocessing

- Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$
- Preprocessing: feature scaling and mean normalization

$$x_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{s_j}$$

  - μ_j: mean
  - s_j: standard deviation

PCA - Algorithm

Reduce data from n-dimensions to k-dimensions

- Compute covariance matrix

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

- Compute eigenvectors of covariance matrix

[U, S, V] = svd(Sigma)

  - U = [$u^{(1)}$, $u^{(2)}$, …, $u^{(n)}$]

PCA - Algorithm

Reduce data from n-dimensions to k-dimensions

- From [U, S, V] = svd(Sigma), we get k first columns
  - $U_{reduce} = [\boldsymbol{u^{(1)}, u^{(2)}, \ldots, u^{(k)}}] \in \mathbb{R}^{nxk}$
- To map $x \in \mathbb{R}^n$ from to $z \in \mathbb{R}^k$

$$z^{(i)} = U^T_{reduce} x^{(i)}$$

(kx1 = kxn x nx1)

Covariance matrix

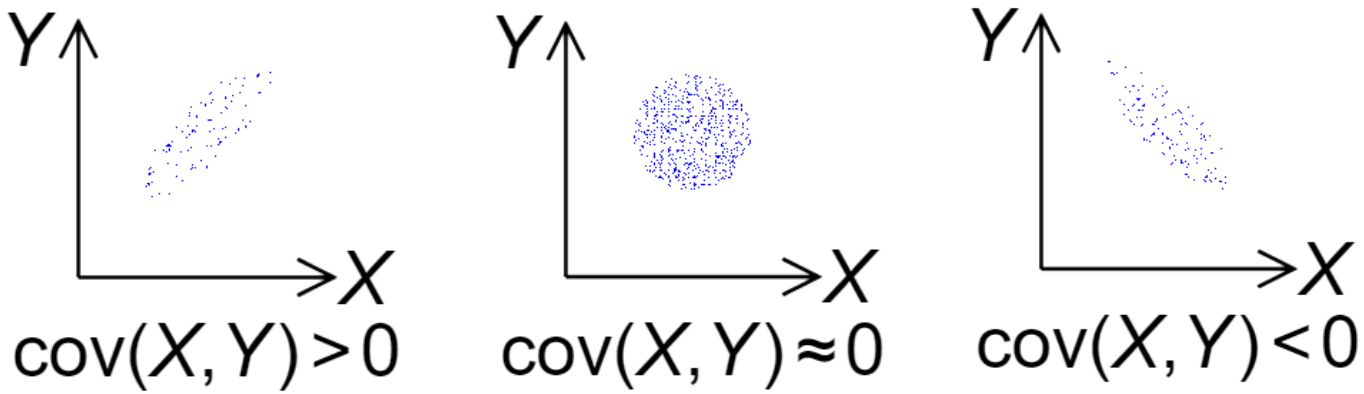
- Represent covariance between dimensions as a matrix, e.g., for 3 dimensions, x, y and z of input A = [x, y, z]

$$C = \begin{vmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{vmatrix}$$

- Diagonal is the variances of x, y and z
- cov(x, y) = cov(y, x) hence matrix is symmetrical about the diagonal
- N-dimensional data will result in NxN covariance matrix

Covariance

- What is the interpretation of covariance calculations?
- For example, 2 dimensional dataset
  - X: number of hours studied for a subject
  - Y: marks obtained in that subject
  - Covariance value is, e.g.,: 104.53
  - What does this value mean?



Source: Wikimedia

Covariance

- Exact value is not as important as its sign
- A positive value of covariance indicates both dimensions increase or decrease together, e.g., as the number of hours studied increases, the marks in that subject increase
- A negative value indicates while one increases the other decreases or vice versa
- If covariance is zero, the two dimensions are independent of each other, e.g., heights of students vs the marks obtained in a subject

Covariance

- Why bother with calculating covariance when we could just plot the two values to see their relationship?
- Covariance calculations are used to find relationships between dimensions in high dimensional data sets (usually greater than 3) where visualization is difficult.

Singular Value Decomposition (SVD)

- Any mxn matrix X can be written as the product of 3 matrices:

$$X = USV^T$$

Where,

- U is mxm and its columns are orthonormal vectors
- V is nxn and its columns are orthonormal vectors
- S is mxn diagonal and its diagonal elements are called the singular values of X
- The columns of U are the eigenvectors of XX^T
- The columns of V are the eigenvectors of X^TX
- The squares of the diagonal elements of S are the eigenvalues of XX^T and X^TX

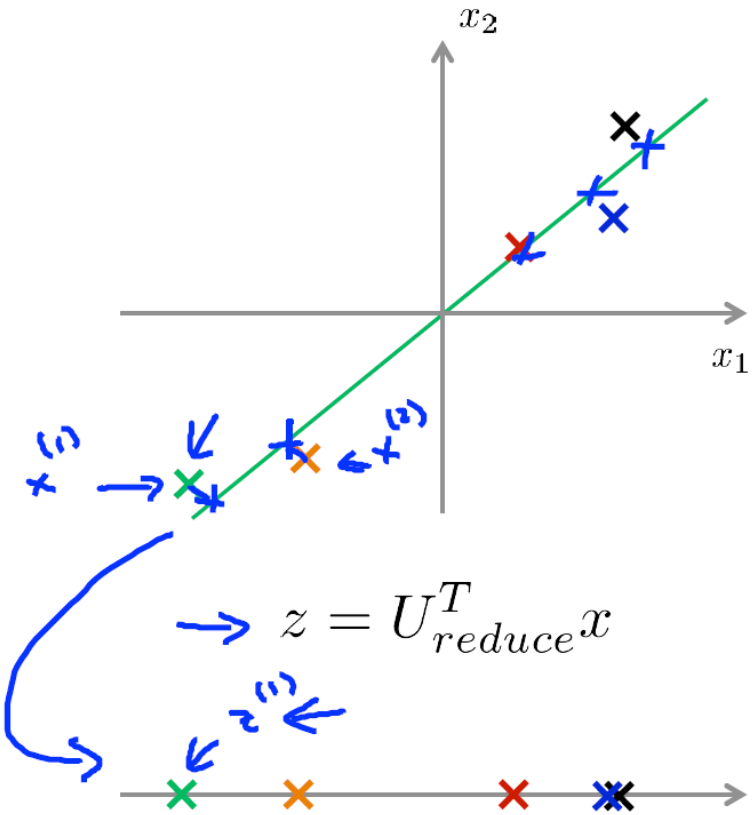
Algorithm – idea of PCA

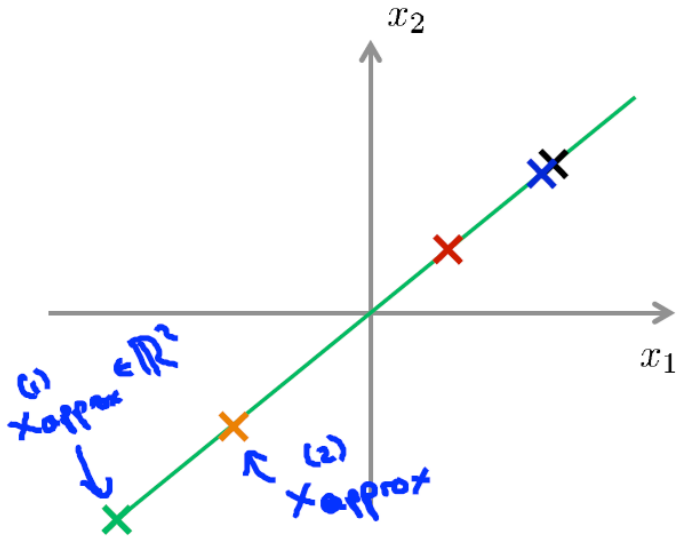
- Given a set of points, how do we know if they can be compressed?
- The answer is to look into the correlation between the points
- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.
- This is the principle component.

Reconstruction

$$x^{(i)}_{approx} = U_{reduce} z^{(i)}$$

(nx1 = nxk x kx1)





Principal component choosing

How to choose k, number of principal components

- Average square error: $\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2$
- Total variance in data: $\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2$
- Typically, choose smallest k so that

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} \leq 0.01 \text{ (1\% error)}$$

99% of variance is retained

Principal component choosing

How to choose k, number of principal components

- Try PCA with k = 1, 2, 3, …
- Compute U_{reduce}, $z^{(1)}$, $z^{(2)}$, …, $z^{(m)}$
- Compute $x^{(1)}_{approx}$, $x^{(2)}_{approx}$,…, $x^{(m)}_{approx}$
- Check if

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x^{(i)}_{approx}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2}\leq 0.01$$

Principal component choosing

How to choose k, number of principal components

[U, S, V] = svd(Sigma)

- For given k, we can approximate the error and choose smallest k so that

$$1 - \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leq 0.01$$

- Or variance retained

$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \geq 0.99$$

Applying PCA to faces

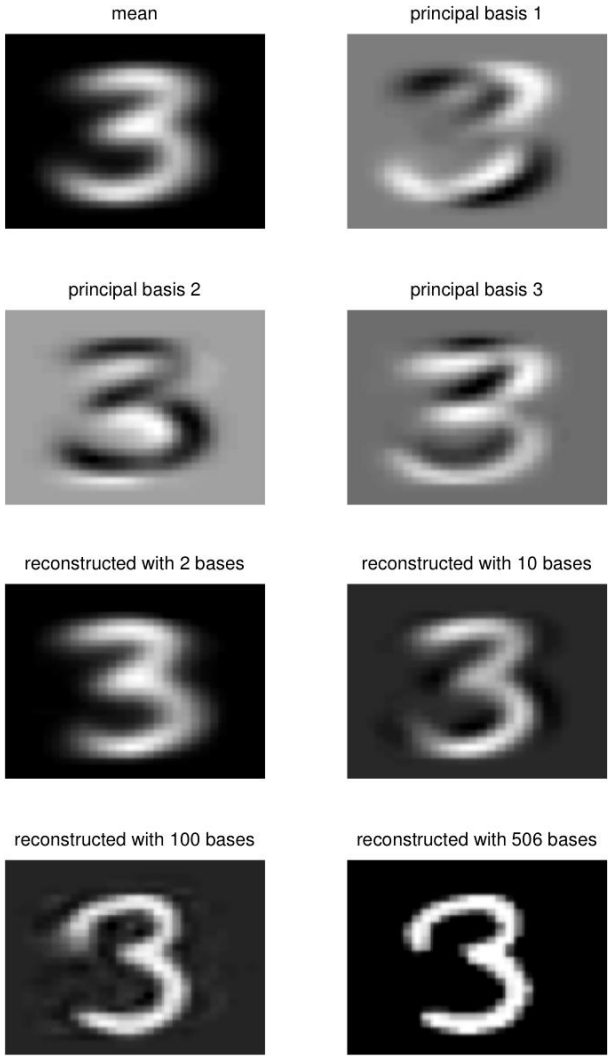
- Consider running PCA on 2429 19x19 grayscale images (CBCL data)
- Can get good reconstructions with only 3 components

- PCA for pre-processing: can apply classifier to latent representation
  - For face recognition PCA with 3 components obtains 79% accuracy on face/non-face discrimination for Gaussian mixture model (GMM).
- Can also be good for visualization

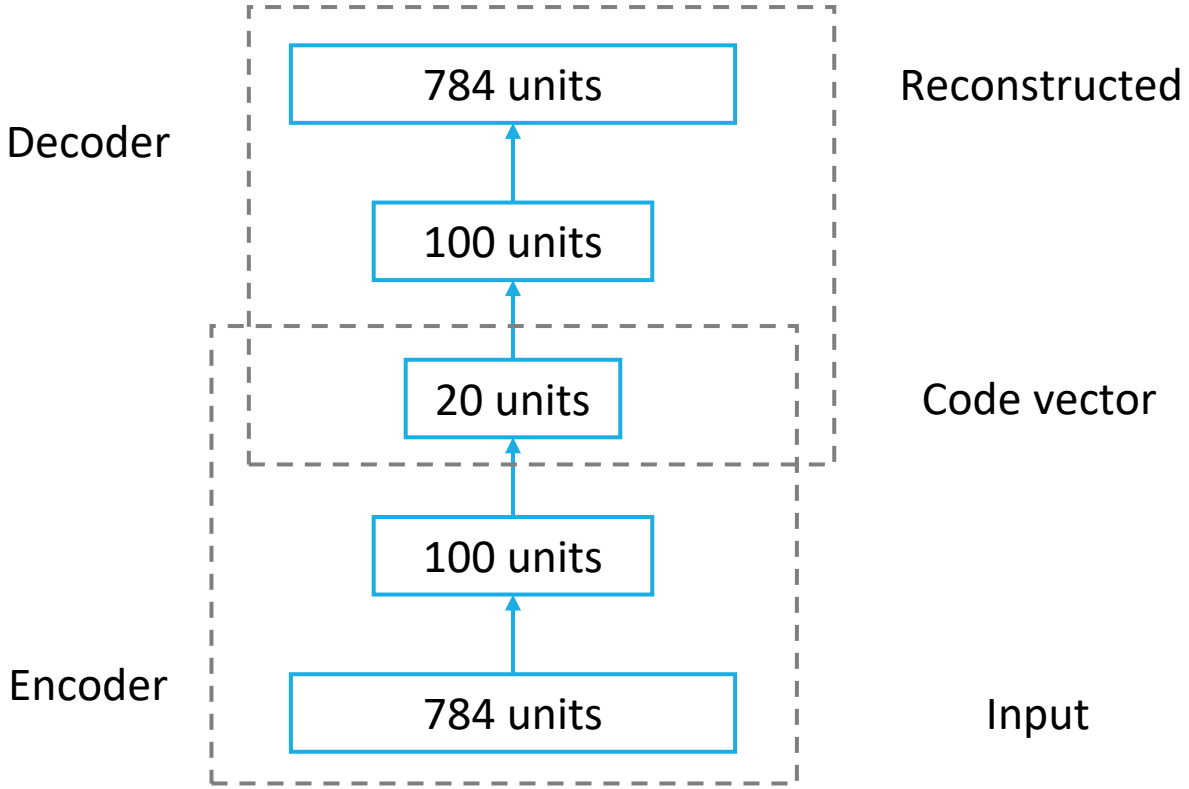
Applying PCA to digits

- Principal components of face images (eigenfaces)

Applying PCA to digits

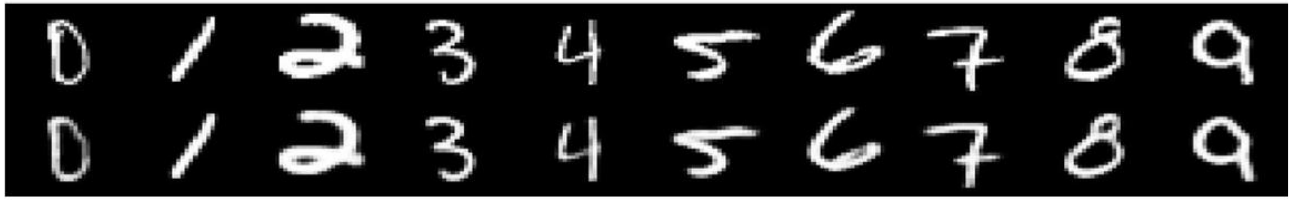


PCA as autoencoder



PCA as autoencoder

- Real data
- Deep encoder (30-D)



- PCA (30-D)

Advice for applying PCA

Supervised learning speed up

- Dataset: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)}), x^{(i)} \in \mathbb{R}^n$
- Run PCA on inputs: $z^{(1)}, z^{(2)}, \ldots, z^{(m)} \in \mathbb{R}^k$
- New training: $(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \ldots, (z^{(m)}, y^{(m)}) \in \mathbb{R}^n$

Mapping x(i) → z(i) should be formed by running PCA only on the training set.

This mapping should be applied to the validation and test sets as well.

Advice for applying PCA

Application of PCA

- Compression
  - Reduce memory/disk needed to store data
  - Speed up learning algorithm
  - Choose k by percent of variance retained
- Visualization
  - Choose k = 2 or k = 3

Advice for applying PCA

Bad use of PCA: to prevent overfitting

- Use $z^{(i)}$ instead of $x^{(i)}$ to reduce the number of features
  - Fewer features, less likely to overfit → bad idea!
- This might work but is not a good way to address overfitting
  - Use regularization instead

Advice for applying PCA

Design of ML system

- Get training set $(x^{(i)}, y^{(i)})$
- Run PCA to reduce dimension of $x^{(i)}$ to get z(i)
- Train model on $(z^{(i)}, y^{(i)})$
- Test model on $z^{(i)}_{test}$

How about doing the whole thing without PCA?

- Before implementing PCA, try first with original data $x^{(i)}$
- Only if that does not work, then implement PCA and use $z^{(i)}$

References

This lecture borrowed ideas and pictures from

- PCA, Andrew Ng
- CSC 411 Lecture 12: Principal Component Analysis, Roger Grosse, Amir-massoud Farahmand, and Juan Carrasquilla, University of Toronto
- Principal Component Analysis, Illinois University
- Principal Components Analysis, The Pennsylvania State University