18AIC301J: DEEP LEARNING TECHNIQUES

B. Tech in ARTIFICIAL INTELLIGENCE, 5th semester

Faculty: **Dr. Athira Nambiar**
Section: A, slot:D
Venue: TP 804
Academic Year: 2022-22

UNIT-2

Limitations of gradient descent learning algorithm, Contour maps,
Momentum based gradient descent, Nesterov accelerated gradient Descent, AdaGrad, RMSProp, Adam learning Algorithm, Stochastic gradient descent
Implement linear regression with stochastic gradient descent
Mini-batch gradient descent, Bias Variance tradeoff, Overfitting in deep neural networks,
Hyperparameter tuning, Regularization: L2 regularization, Dataset Augmentation and Early stopping
Implement linear regression with stochastic mini-batch gradient descent and compare the results with previous exercise.
Dimensionality reduction, Principal Component Analysis, Singular value decomposition
Autoencoders, Relation between PCA and Autoencoders, Regularization in Autoencoders
Optimizing neural networks using L2 regularization, Dropout, data augmentation and early stopping.

Dimensionality reduction

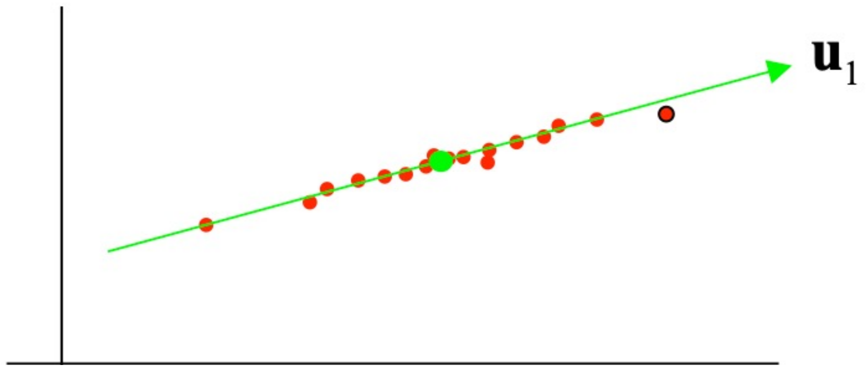
Usage

- Have data of dimension d
- Reduce dimensionality to k<d
  - Discard unimportant features
  - Combine several features in one
- Use resulting k-dimensional data set for
  - Learning for classification problem (e.g. parameters of probabilities P(x|C)
  - Learning for regression problem (e.g. parameters for model y=g(x|Thetha)

Dimensionality reduction

Subspace Models

- Often in machine learning problems, input vectors have high dimensionality *D*
  - for an 1800 x 1200 colour image, *D*≅6.5 million.
  - for a 1-second acoustic voice signal sampled at 5kHz, *D* = 5,000
- There is typically insufficient training data to learn a probabilistic model in such a high-dimensional space.
- Fortunately, these signals usually live in a much smaller subspace, or manifold, of this high-dimensional space.

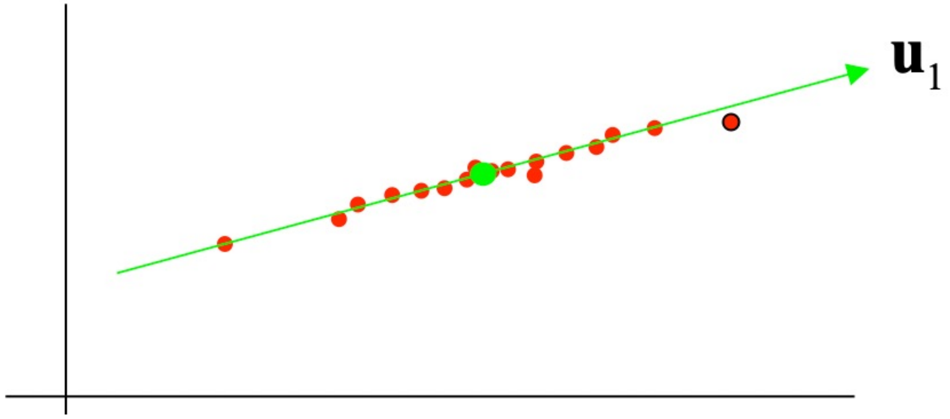


Dimensionality reduction

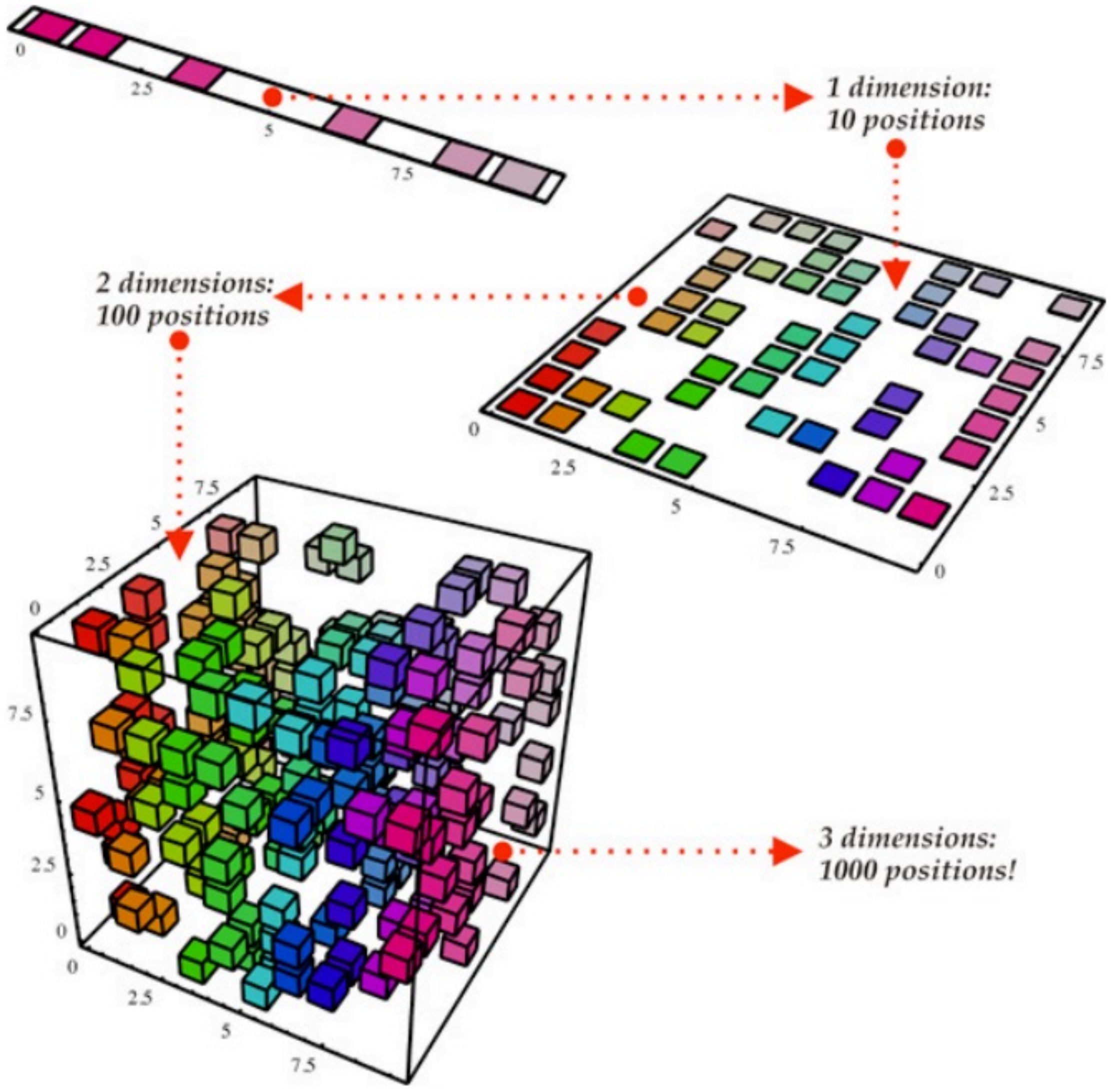
Subspace Models

Probability & Bayesian Inference

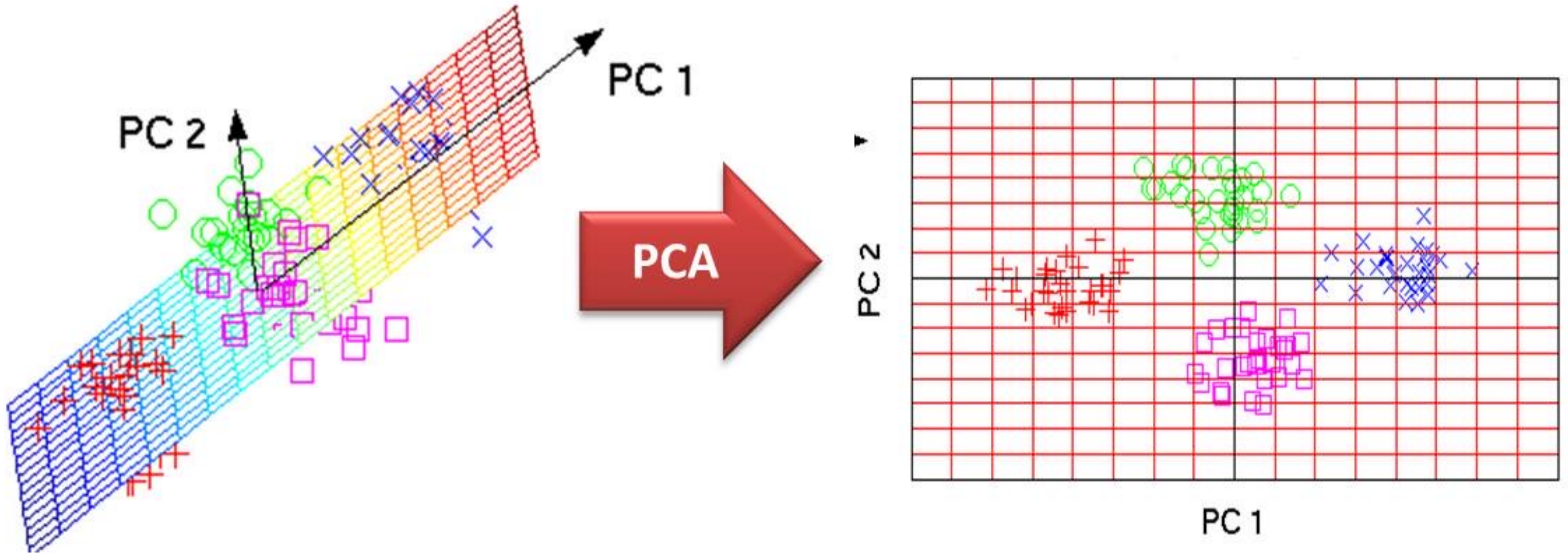
- The goal of subspace methods is to discover the low-dimensional subspace in which the data lie and exploit the lower-dimensionality to allow efficient and detailed modeling.



Dimensionality reduction

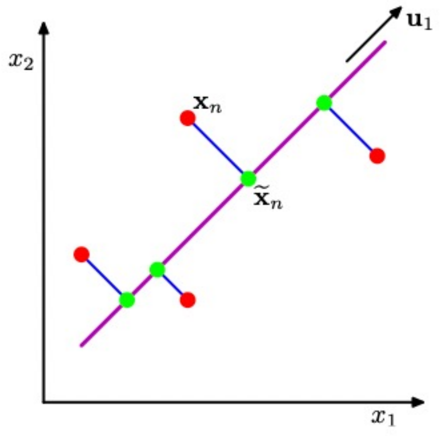


Principal Component Analysis



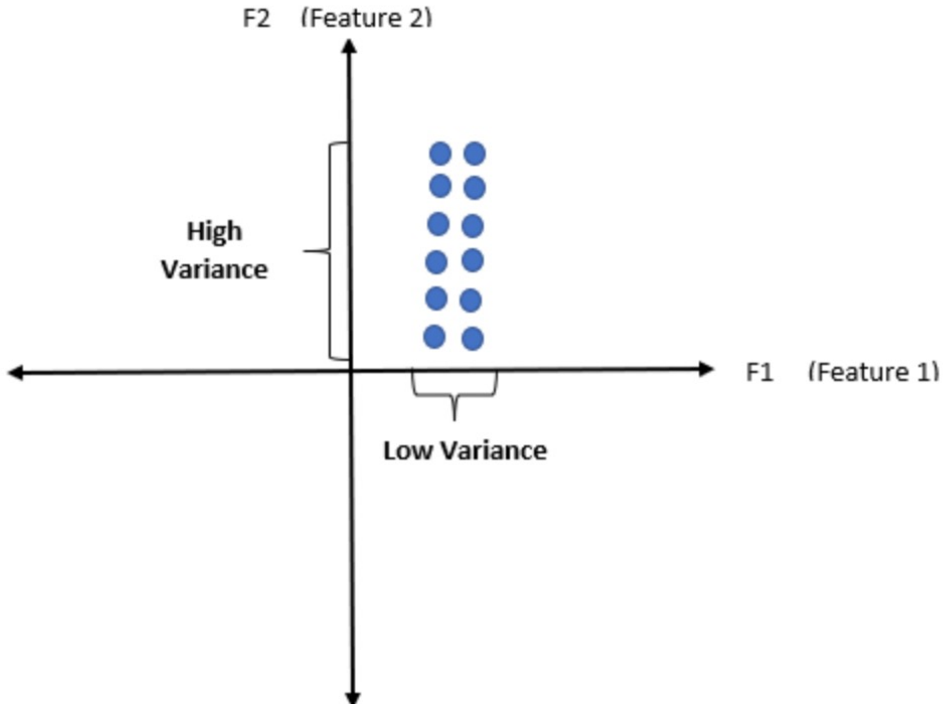
Principal Component Analysis

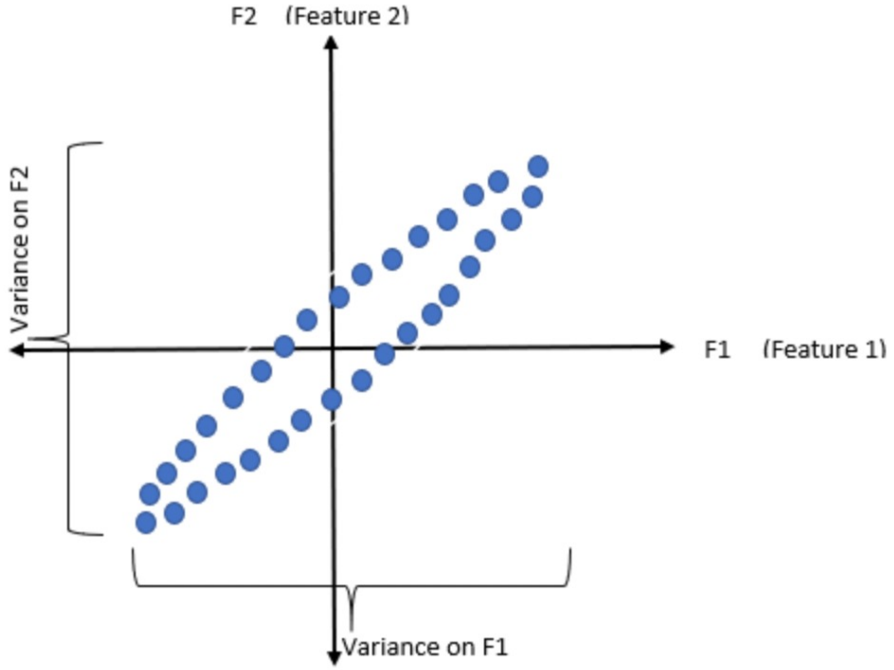
- PCA finds the linear subspace that
  - maximizes the explained variance
  - equivalently, minimizes the unexplained variance
- PCA can be applied to any multidimensional dataset
  - (data do not have to be Gaussian)



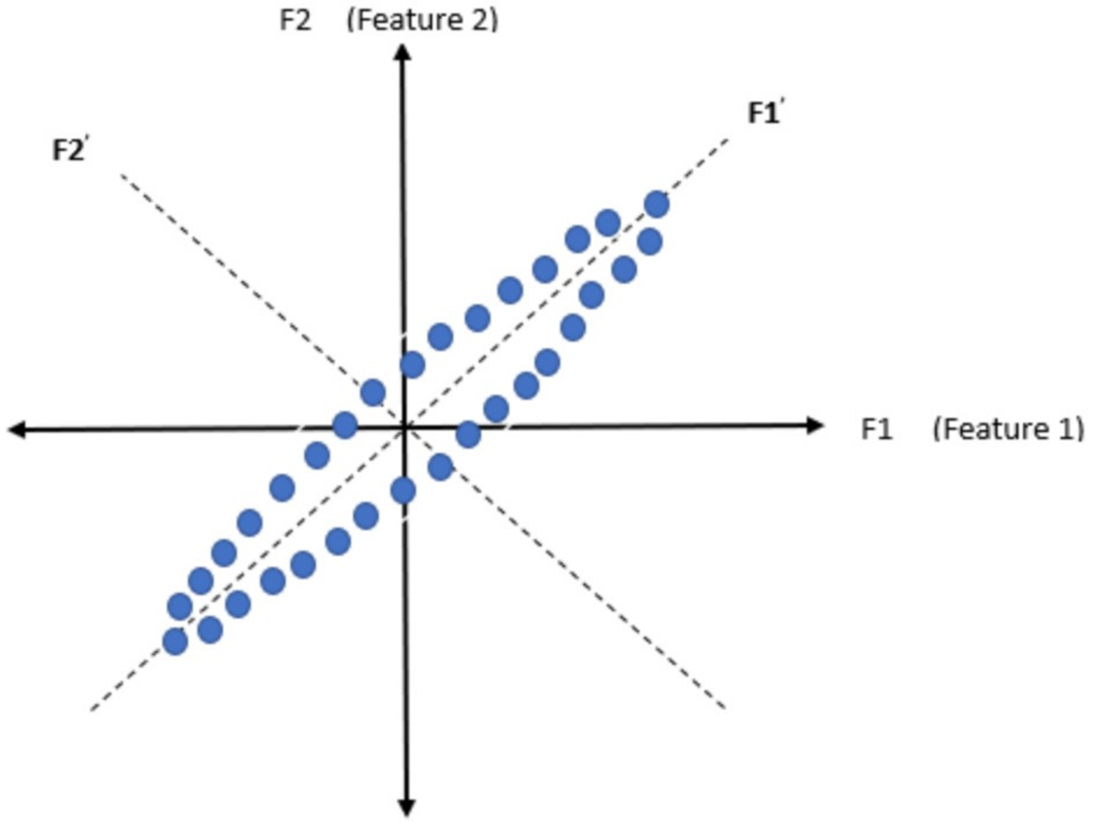
PCA Intuition.

PCA is a linear dimensionality reduction technique which converts a set of correlated features in the high dimensional space into a series of uncorrelated features in the low dimensional space. These uncorrelated features are also called principal components. PCA is an orthogonal linear transformation which means that all the principal components are perpendicular to each other. It transforms the data in such a way that the first component tries to explain maximum variance from the original data. It is an unsupervised algorithm i.e. it does not take into consideration the class labels.

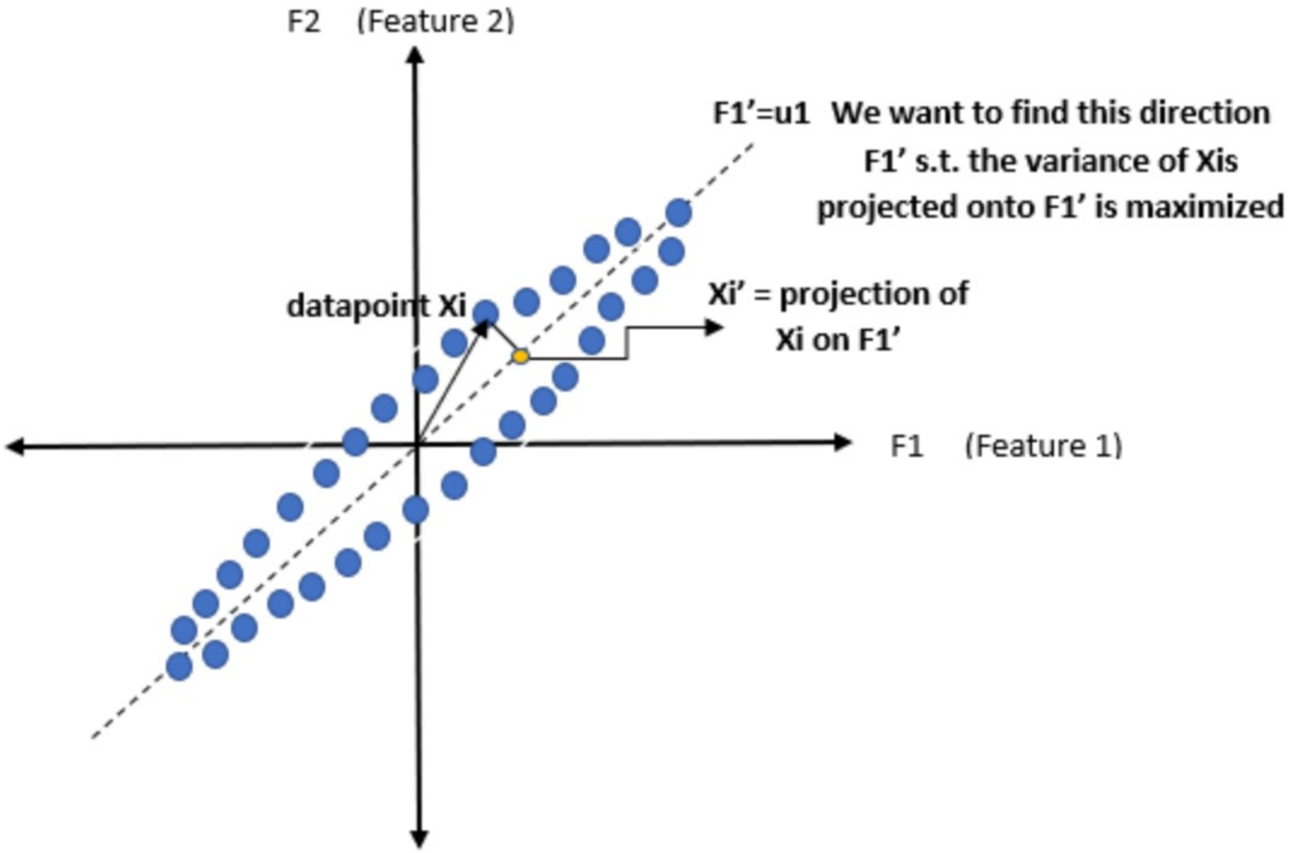

F2 (Feature 2)
High Variance
F1 (Feature 1)
Low Variance


F2 (Feature 2)
Variance on F2
F1 (Feature 1)
Variance on F1

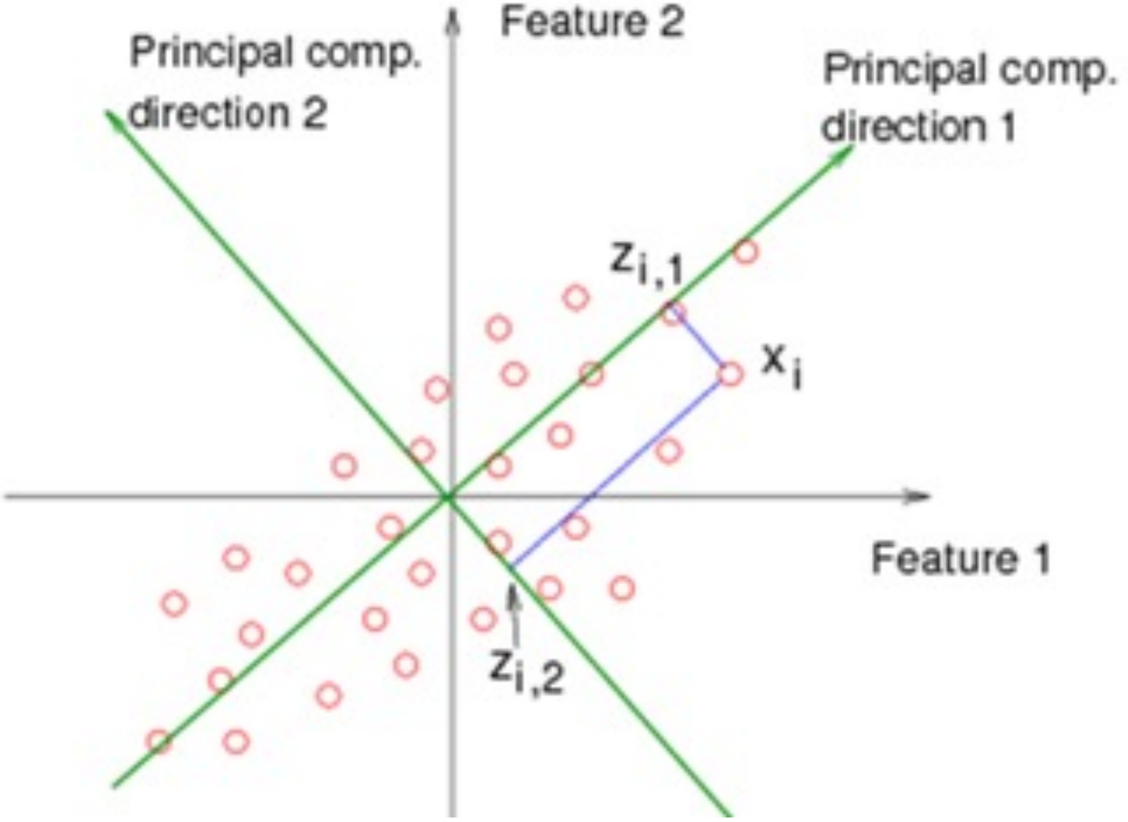
Here, PCA comes into the picture

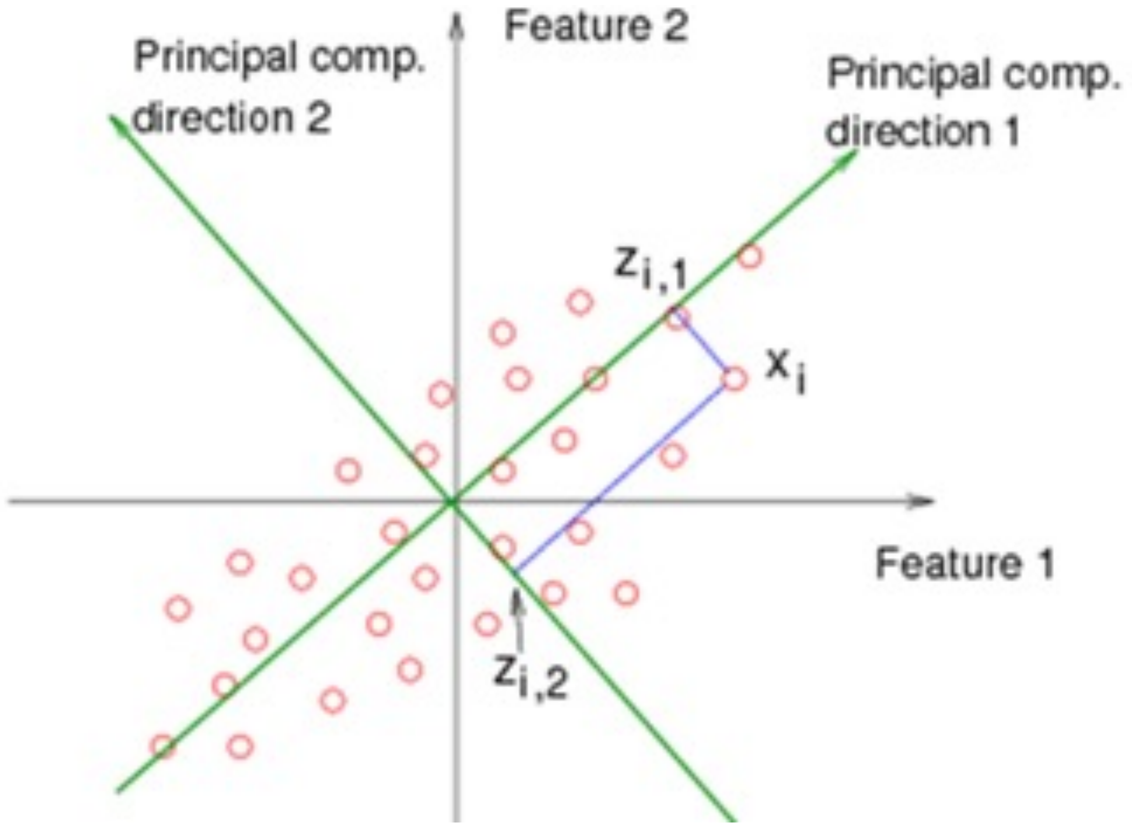


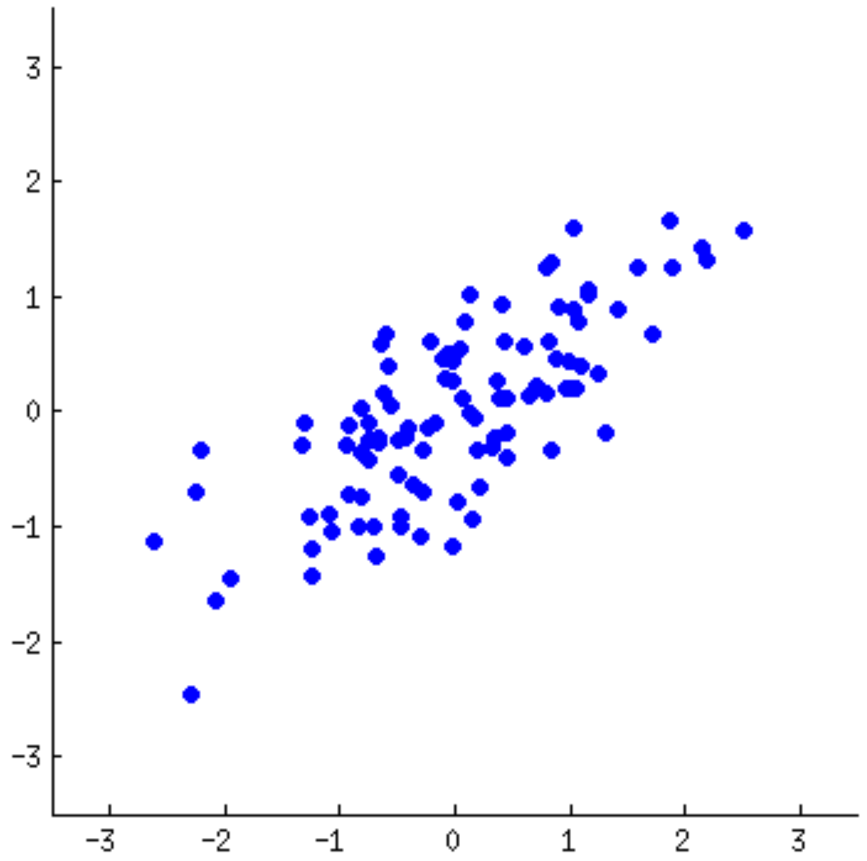
PCA performs linear orthogonal transformation on the data to find features F1' and F2' such that the variance on F1' >> variance on F2'.


F2 (Feature 2)
F1'=u1 We want to find this direction F1' s.t. the variance of Xis projected onto F1' is maximized
datapoint Xi
Xi' = projection of Xi on F1'
F1 (Feature 1)

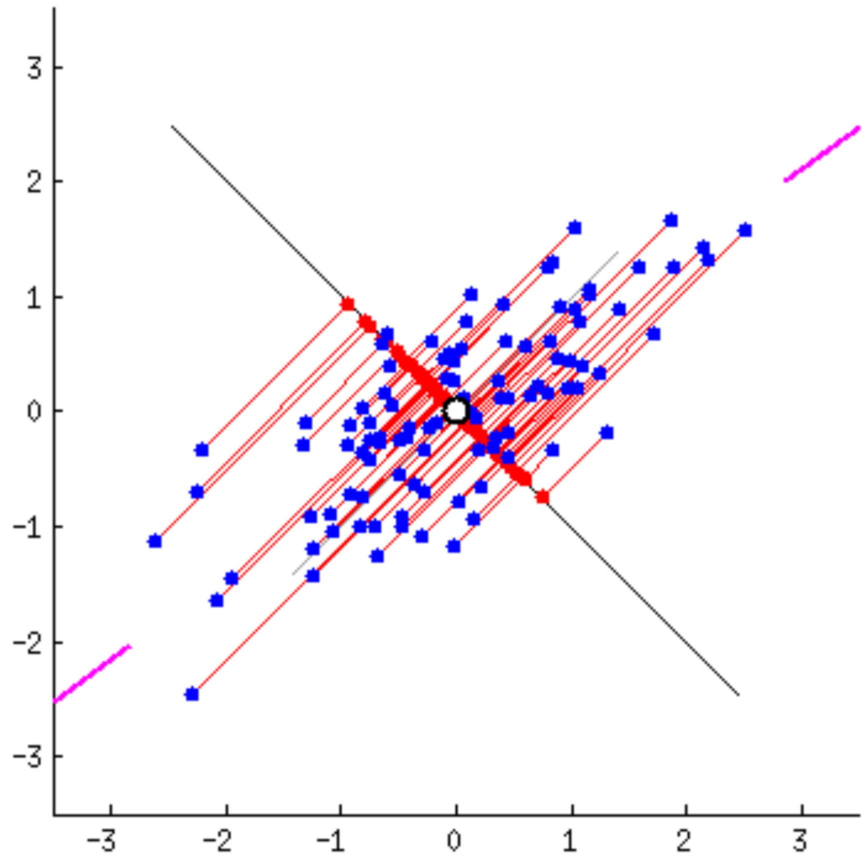
PCA tries to find a direction F1' such that the variance of point Xis projected onto F1' is maximized.


Feature 2
Principal comp. direction 2
Principal comp. direction 1
$z_{i,1}$
x_i
Feature 1
$z_{i,2}$


Feature 2
Principal comp. direction 2
Principal comp. direction 1
$z_{i,1}$
x_i
Feature 1
$z_{i,2}$



3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3



3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3

Principal Component Analysis

- Choose directions such that a total variance of data will be maximum
  - Maximize Total Variance

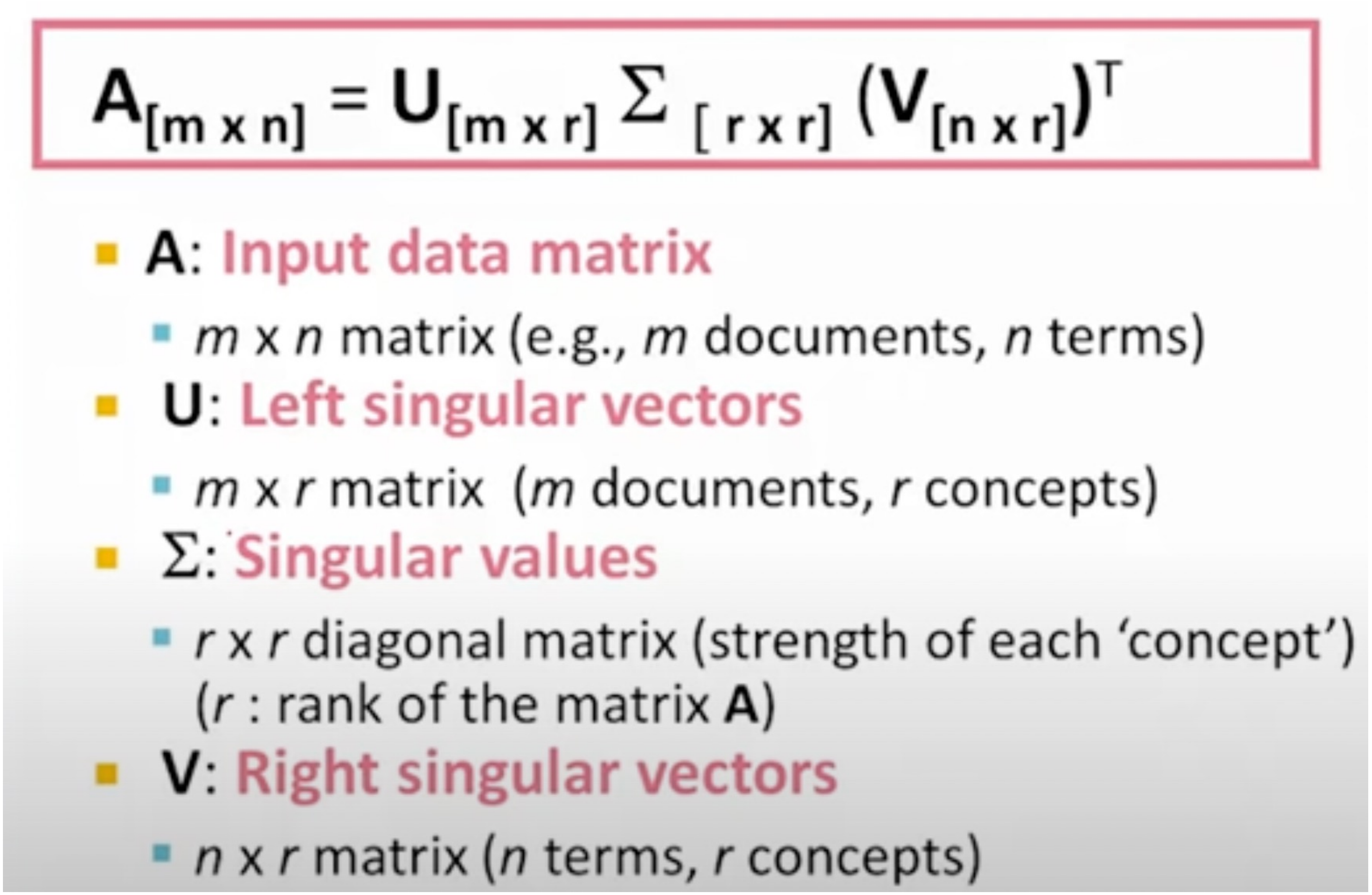
- Choose directions that are orthogonal
  - Minimize correlation

- Choose $k<d$ orthogonal directions which maximize total variance

PCA

- d-dimensional feature space
- d by d symmetric covariance matrix estimated from samples $\mathrm{Cov}(x) = \Sigma,$
- Select k largest eigenvalue of the covariance matrix and associated k eigenvectors
- The first eigenvector will be a direction with largest variance

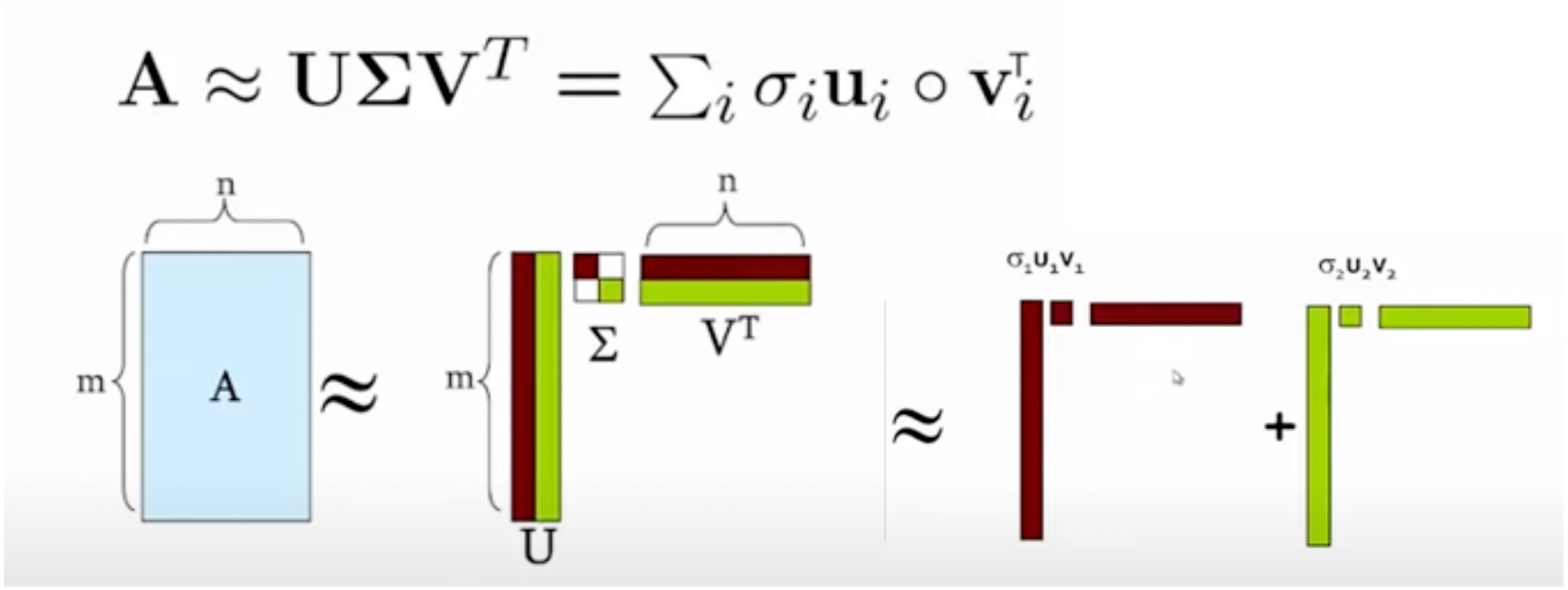
Singular value decomposition



$$A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T$$

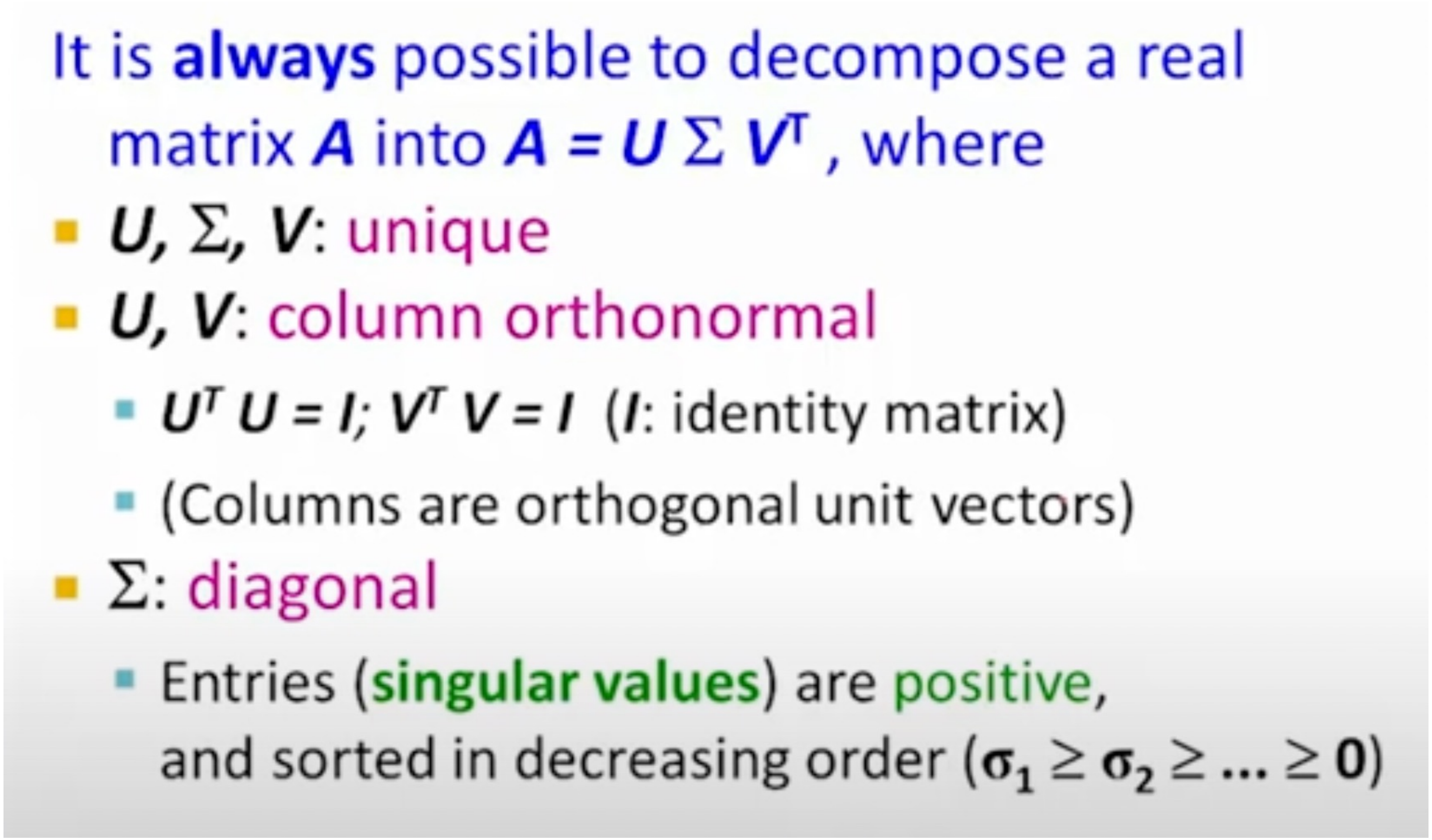
- A: Input data matrix
  - *m* x *n* matrix (e.g., *m* documents, *n* terms)
- U: Left singular vectors
  - *m* x *r* matrix (*m* documents, *r* concepts)
- Σ: Singular values
  - *r* x *r* diagonal matrix (strength of each 'concept') (*r* : rank of the matrix **A**)
- V: Right singular vectors
  - *n* x *r* matrix (*n* terms, *r* concepts)

Singular value decomposition



$$\mathbf{A} \approx \mathbf{U}\Sigma\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$

Singular value decomposition



It is **always** possible to decompose a real matrix ***A*** into ***A* = *U* Σ *V*^{T}**, where

- ***U*, Σ, *V***: unique
- ***U*, *V***: column orthonormal
  - $U^T U = I$; $V^T V = I$ (***I***: identity matrix)
  - (Columns are orthogonal unit vectors)
- Σ: diagonal
  - Entries (**singular values**) are positive, and sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$)

Learning Resources

- Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.
- Eugene Charniak, Introduction to Deep Learning, MIT Press, 2018.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
- Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
- Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
- https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Thank you