Machine Learning Algorithms: A Mathematical perspective

Linear Algebra for Machine Learning

Amir Hosein Hadian

February 3, 2022

Outline

Course introduction

What will you learn during this course?

- Linear algebra for machine learning:
  - Basic algebra and algebraic definition of vector space
  - Linear transformation and its applications
  - Matrix operations
  - Special Matrices
  - Foundations of numpy
  - Eigenvalue and eigenvector
  - Eigenvalue decomposition
  - Singular value decomposition
  - Orthogonality and analytical geometry
  - Gram-Schmidt algorithm
  - LU decomposition
  - QR decomposition
  - Cholesky decomposition
  - Multivariate statistics through matrices
  - Principle component analysis (PCA)
  - Linear discriminant analysis (LDA)

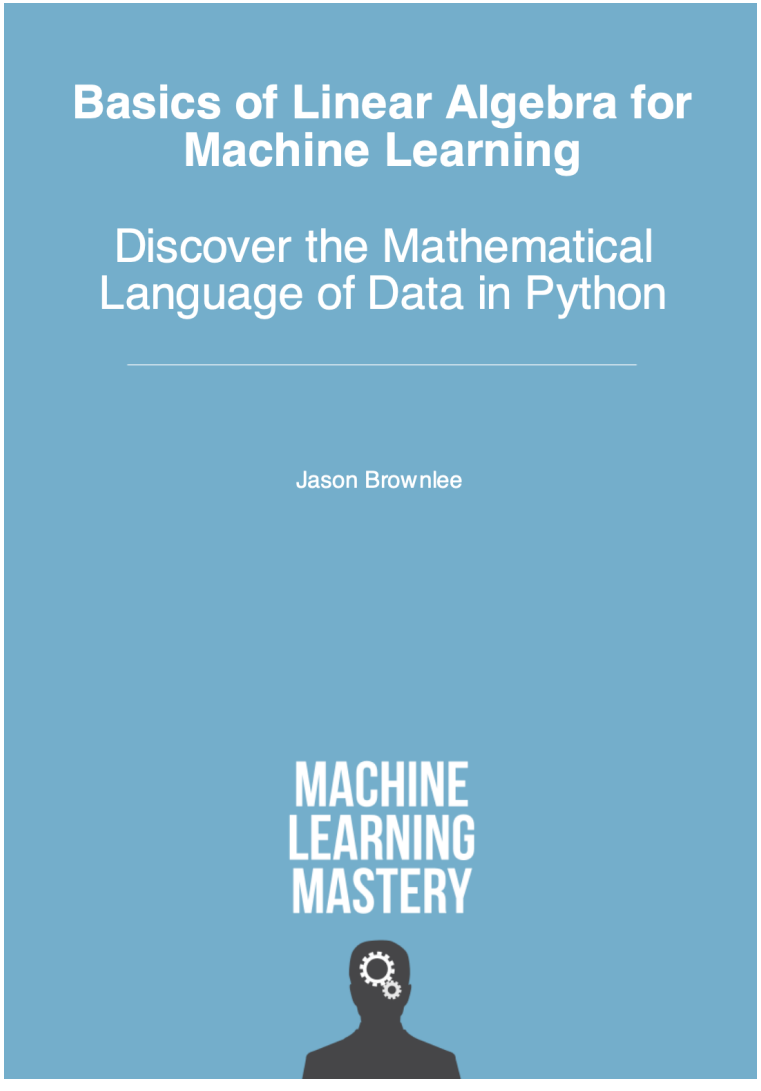
What will you learn during this course?

- Probability for machine learning:
  - Discrete and continuous probability functions
  - Sum rule, product rule, and Bayes' theorem
  - Summary statistics and independence
  - Classification using Naive Bayes algorithm
  - Gaussian mixture model
  - Parameter learning via maximum likelihood
  - EM algorithm

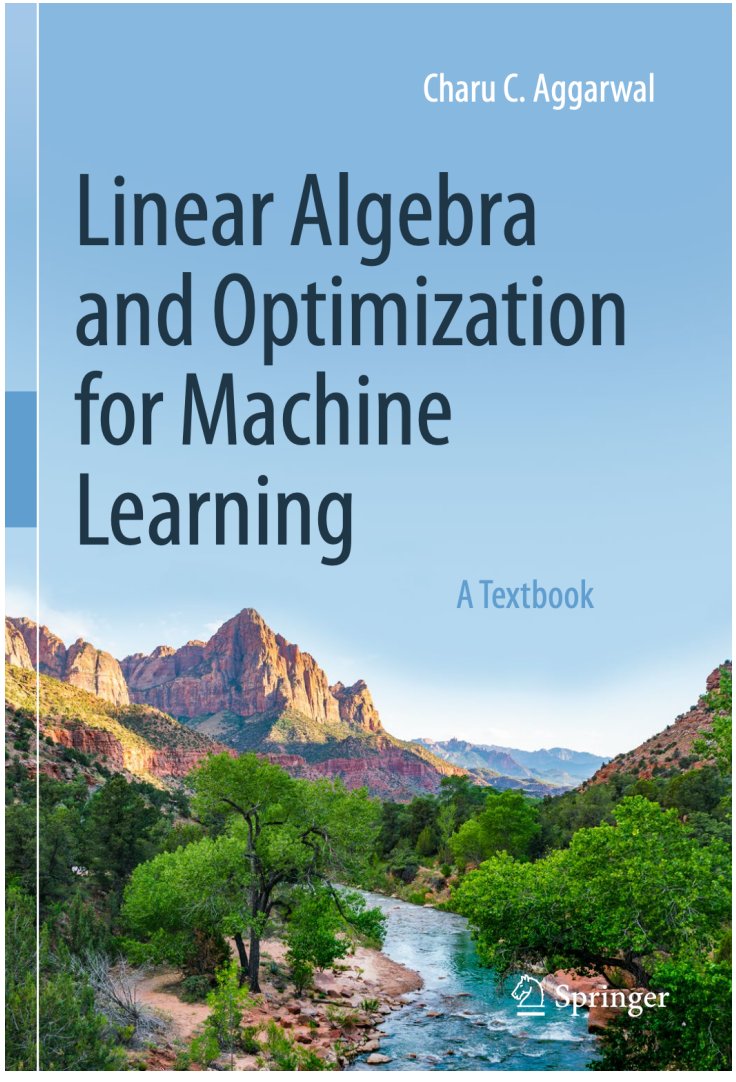
What will you learn during this course?

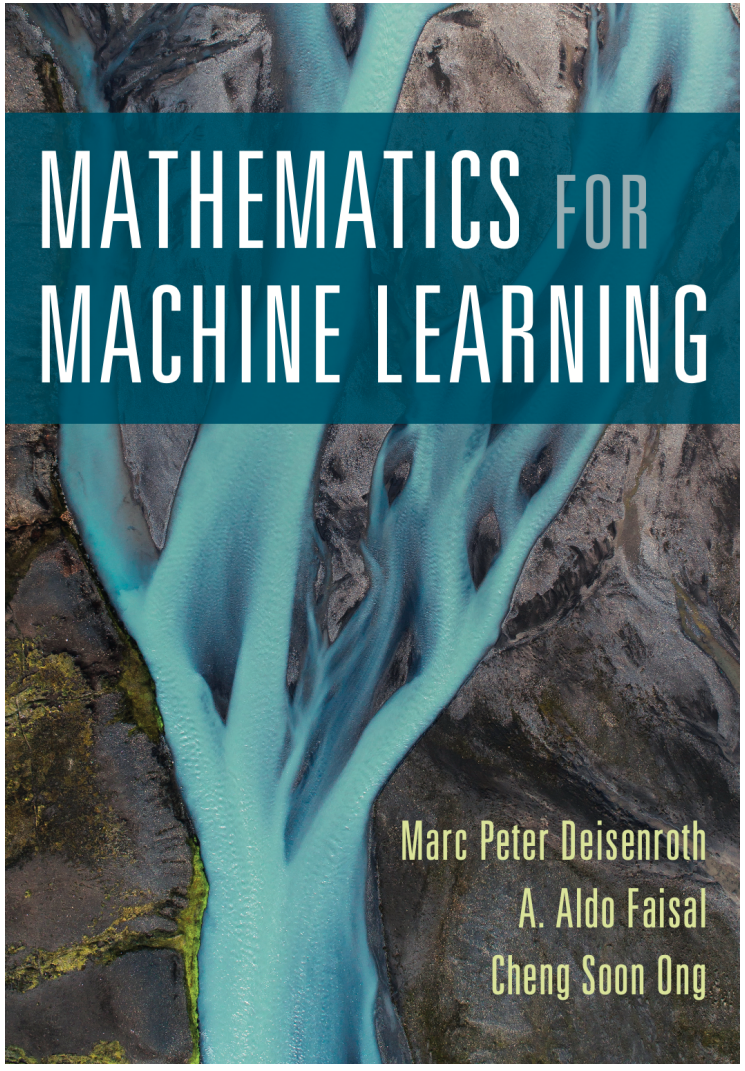
- Optimization for machine learning:
  - Optimization Using Gradient Descent
  - Constrained Optimization and Lagrange Multipliers
  - Convex Optimization
  - Support Vector Machines (SVM)
  - Recommender system using matrix decomposition

References

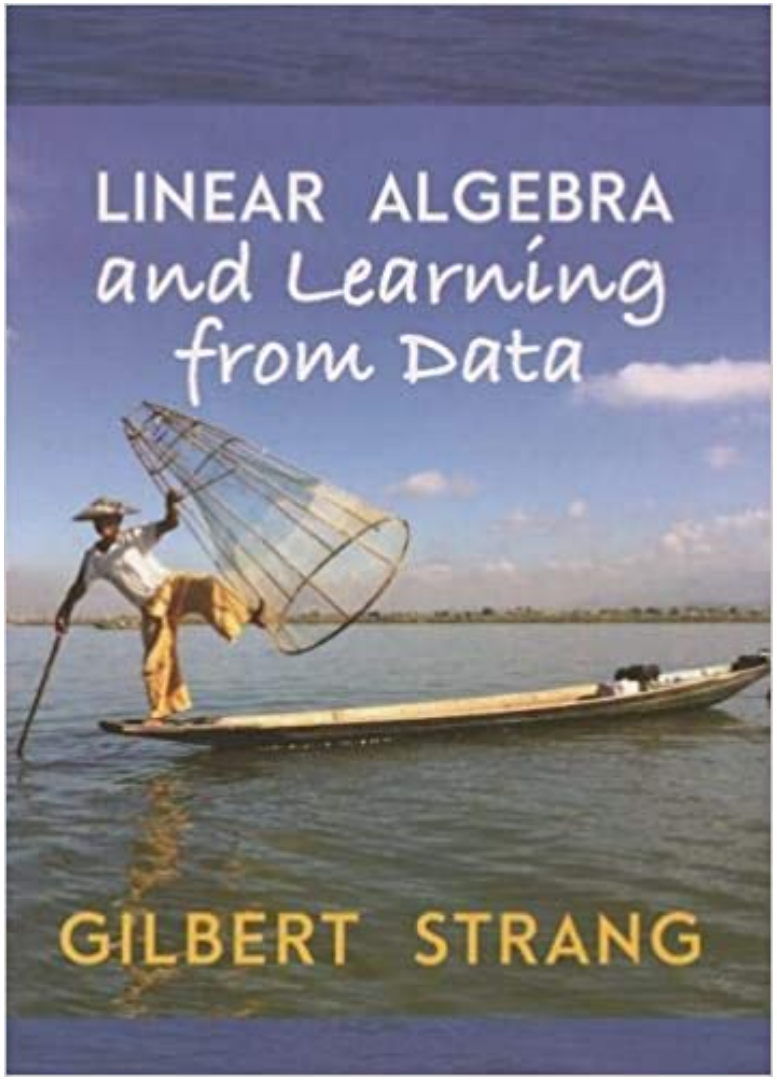

Basics of Linear Algebra for Machine Learning
Discover the Mathematical Language of Data in Python
Jason Brownlee
MACHINE LEARNING MASTERY

References

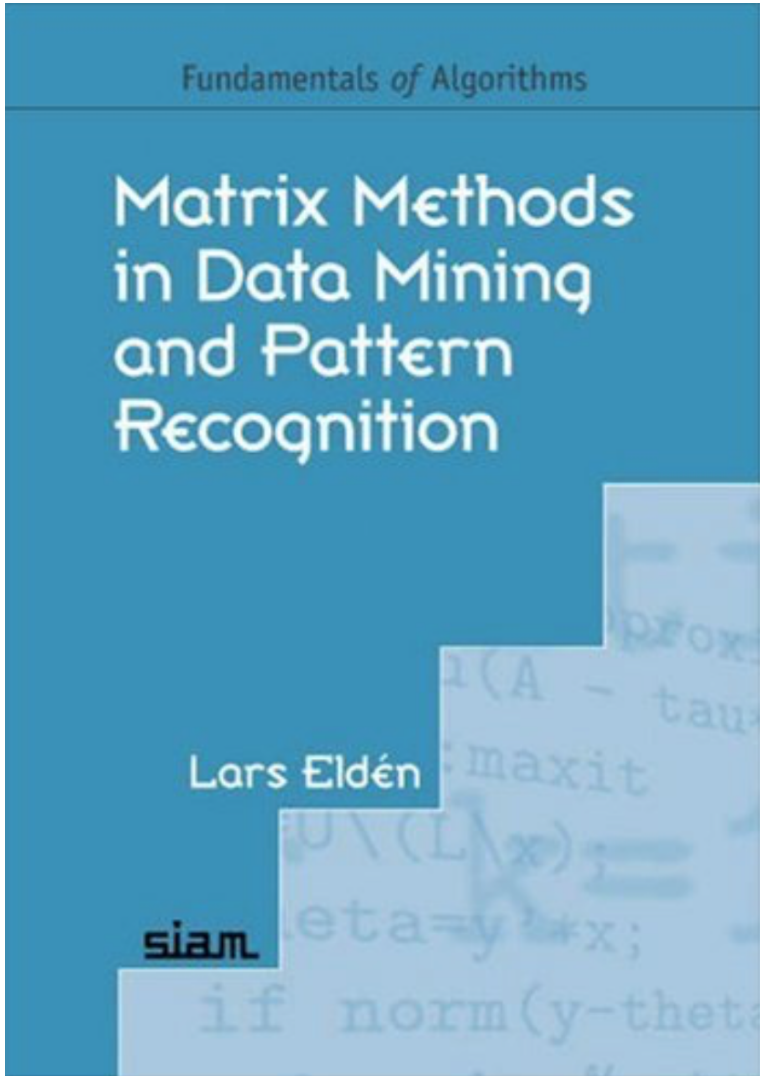




References



References



Motivation

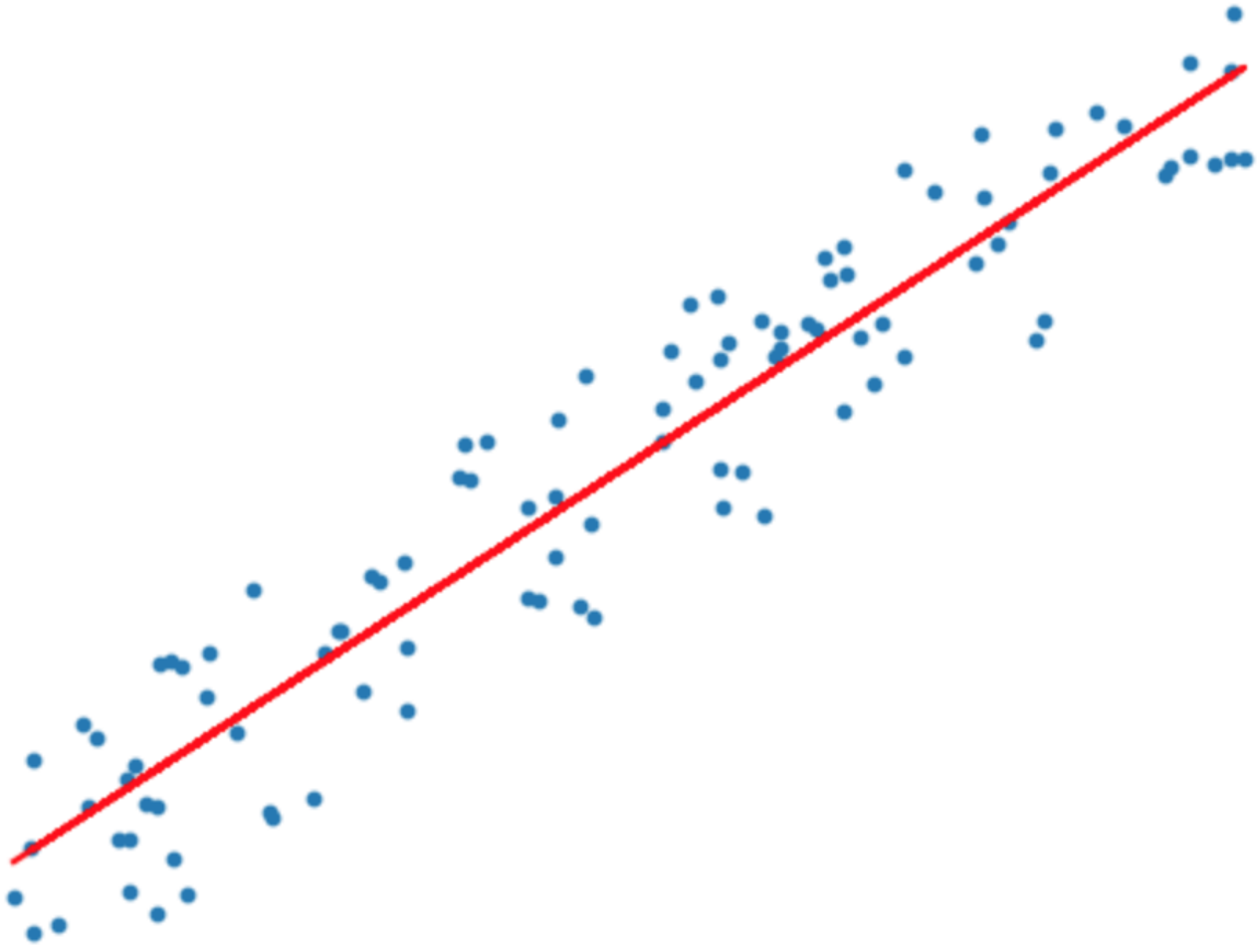
Reasons to NOT Learn Linear Algebra

- It's not required
- It's slow
- It's a huge field

Dataset and Data Files

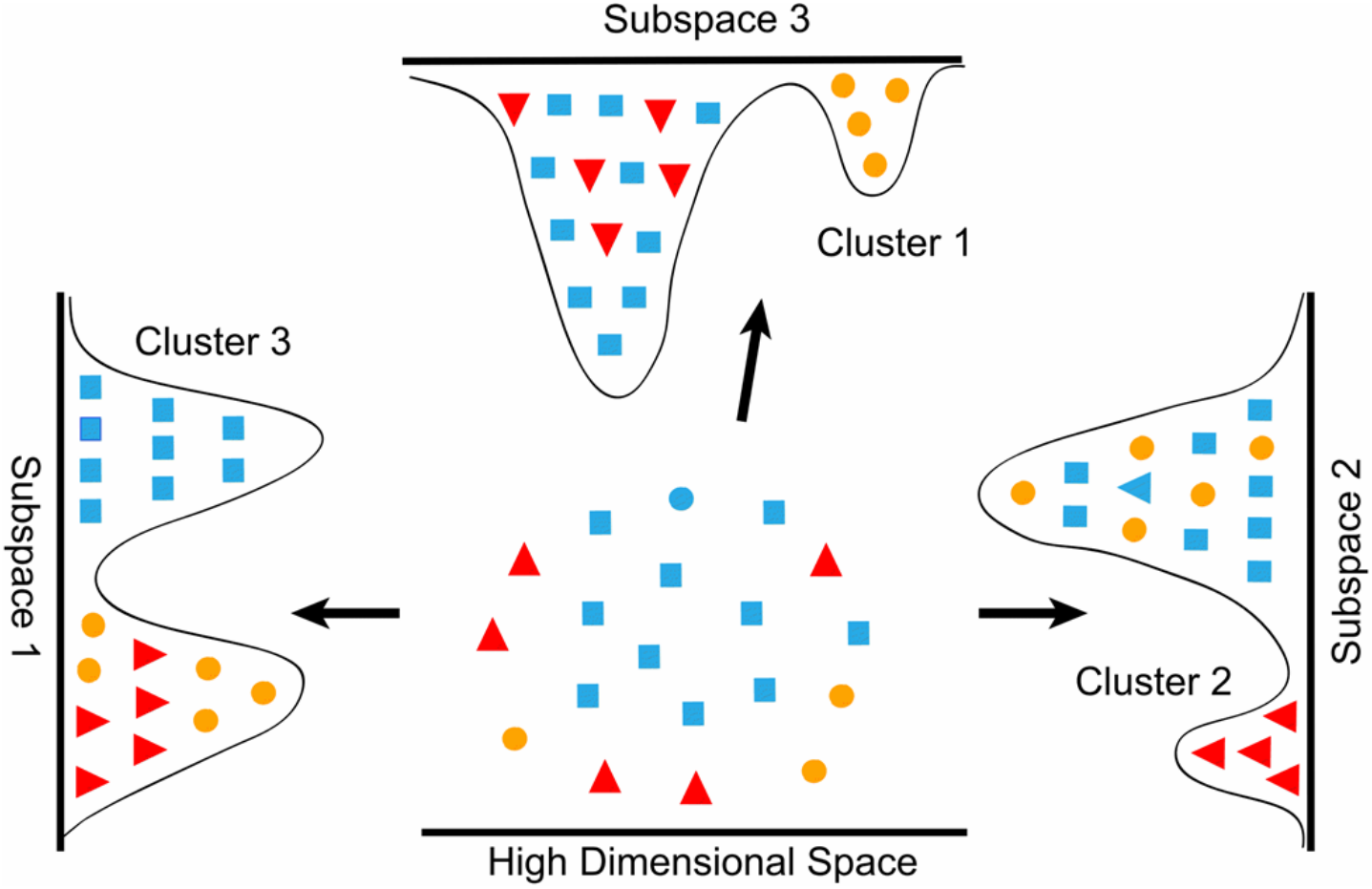
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.9	3	4.2	1.5	Iris-versicolor
6	6	2.2	4	1	Iris-versicolor
7	6.1	2.9	4.7	1.4	Iris-versicolor
8	5.6	2.9	3.6	1.3	Iris-versicolor
9	6.1	3	4.9	1.8	Iris-virginica
10	6.4	2.8	5.6	2.1	Iris-virginica
11	7.2	3	5.8	1.6	Iris-virginica
12	7.4	2.8	6.1	1.9	Iris-virginica
13	7.9	3.8	6.4	2	Iris-virginica
14	6.4	2.8	5.6	2.2	Iris-virginica

Linear Regression



$$\min_{\theta} \frac{1}{N} \|y - \mathbf{X}\theta\|^2 + \lambda \|\theta\|^2$$

Dimension reduction



$$f : R^n \longrightarrow R^m$$

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
Web	0	1	1	0	0

$$
\begin{array}{r}
 \\
\text{Star Wars} \\
\text{Blade Runner} \\
\text{Amelie} \\
\text{Delicatessen}
\end{array}
\begin{array}{c}
\begin{array}{ccc} \text{Ali} & \text{Beatrix} & \text{Chandra} \end{array} \\
\begin{bmatrix} 5 & 4 & 1 \\ 5 & 5 & 0 \\ 0 & 0 & 5 \\ 1 & 0 & 4 \end{bmatrix}
\end{array}
=
\begin{bmatrix} -0.6710 & 0.0236 & 0.4647 & -0.5774 \\ -0.7197 & 0.2054 & -0.4759 & 0.4619 \\ -0.0939 & -0.7705 & -0.5268 & -0.3464 \\ -0.1515 & -0.6030 & 0.5293 & -0.5774 \end{bmatrix}
$$

$$
\begin{bmatrix} 9.6438 & 0 & 0 \\ 0 & 6.3639 & 0 \\ 0 & 0 & 0.7056 \\ 0 & 0 & 0 \end{bmatrix}
$$

$$
\begin{bmatrix} -0.7367 & -0.6515 & -0.1811 \\ 0.0852 & 0.1762 & -0.9807 \\ 0.6708 & -0.7379 & -0.0743 \end{bmatrix}
$$

Introduction to Vector Spaces

Binary operation and its properties

A binary operation $*$ on a nonempty set S is a function from $S \times S$ to S

$$* : S \times S \longrightarrow S$$

- Addition, subtraction, multiplication are binary operations on Z.
- Addition is a binary operation on Q because ...
- Division is NOT a binary operation on Z because ...

Binary operation and its properties

- $*$ is associative if :

$$\forall a, b, c \in S \quad a * (b * c) = (a * b) * c$$

- $*$ is commutative if :

$$\forall a, b \in S \quad a * b = b * a$$

- $e \in S$ is identity element of S w.r.t. $*$ if:

$$\forall a \in S \quad a * e = e * a = a$$

- a is invertible w.r.t. $*$ if there exists $b \in S$ if:

$$a * b = b * a = e$$

Groupoid or Magma

A Groupoid is a basic kind of algebraic structure that consists of a set equipped with a single binary operation that must be closed by definition. No other properties are imposed.

$$(S; *)$$

Semi-Group

A semi-group is an algebraic structure consisting of a set together with an associative binary operation.

$$(S; *)$$

- $*$ is associative: $\forall a, b, c \in S \quad a * (b * c) = (a * b) * c$

Monoid

A Monoid is an algebraic structure with a single associative binary operation and an identity element.

$$(S; *, e)$$

- $*$ is associative: $\forall a, b, c \in S \quad a * (b * c) = (a * b) * c$
- $e \in S$ is identity element of S w.r.t. $*$:
  $\forall a \in S \quad a * e = e * a = a$

Group

A Group is a set equipped with a binary operation that combines any two elements to form a third element in such a way that three conditions called group axioms are satisfied, namely associativity, identity and invertibility.

$$(S; *, e)$$

- $*$ is associative: $\forall a, b, c \in S \quad a * (b * c) = (a * b) * c$
- $e \in S$ is identity element of S w.r.t. $*$:
  $\forall a \in S \quad a * e = e * a = a$
- Each elements of S are invertible

Abelian Group

An Abelian Group, is a group in which the result of applying the group operation to two group elements does not depend on the order in which they are written. That is, the group operation is commutative:

$$(S; *, e)$$

- $*$ is commutative: $\forall a, b \in S \quad a * b = b * a$
- $*$ is associative: $\forall a, b, c \in S \quad a * (b * c) = (a * b) * c$
- $e \in S$ is identity element of S w.r.t. $*$:
  $\forall a \in S \quad a * e = e * a = a$
- Each elements of S are invertible

Ring

A Ring is a set R equipped with two binary operations satisfying the following axioms:

$$(R, +, \cdot)$$

- $(R, +)$ is an Abelian Group
- $(R, \cdot)$ is a Monoid

A Field is a set F equipped with two binary operations satisfying the following axioms:

$$(F, +, \cdot)$$

- $(F, +)$ is an Abelian Group
- $(F - \{0\}, \cdot)$ is an Abelian Group

A vector space V is a set that is closed under finite vector addition and scalar multiplication. The scalars are members of a field F, in which case V is called a vector space over F.

$$(V, +, \cdot)$$

- $(V, +)$ is Abelian Goroup
- $\lambda(x + y) = \lambda x + \lambda y$
- $\lambda(\psi x) = (\lambda\psi)x$
- $(\lambda + \psi)x = \lambda x + \psi x$

Linear Combination

Consider a vector space V and a finite number of vectors $x_1, \cdots, x_k \in V$. Then every $v \in V$ of the following form:

$$v = \lambda_1 x_1 + \cdots + \lambda_k x_k = \sum_{i=1}^{k} \lambda_i x_i \in V$$

with $\lambda_1, \cdots, \lambda_k \in R$ is a linear combination of the vectors $x_1, \cdots, x_k$

Linear Independence

Finite vectors $x_1, \cdots, x_k \in V$ are called linear independent if:

$$\sum_{i=1}^{k} \lambda_i x_i = 0 \Longleftrightarrow \lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$$

Consider a vector space $V = (V, +, \cdot)$ and set of vectors $A = \{x_1, ..., x_k\} \subseteq V$. If every vector $v \in V$ can be expressed as a linear combination of $x_1, \cdots, x_k$, A is called a generating set of V . The set of all linear combinations of vectors in A is called the span of A. If A spans the vector space V , we write $V = \text{span}[A]$ or $V = \text{span}[x_1, \cdots, x_k]$.

Consider a vector space $V = (V, +, \cdot)$ and $A \subseteq V$. A generating set A of V is called minimal if there exists no smaller set that spans V. Every linearly independent generating set of V is minimal and is called a basis of V .

For vector spaces V, W , a mapping $\Phi : V \longrightarrow W$ is called a linear mapping (or vector space homomorphism/ linear transformation) if:

$$\forall x, y \in V \forall \lambda \in R : \Phi(x + \lambda y) = \Phi(x) + \lambda \Phi(y)$$

Coordinates

Consider a vector space V and an ordered basis $B = (b_1, ..., b_n)$ of V.For any $x \in V$ we obtain a unique representation (linear combination):

$$x = \alpha_1 b_1 + \cdots + \alpha_n b_n$$

of x with respect to B. Then $\alpha_1, \cdots, \alpha_n$ are the coordinates of x with respect to B.

Transformation Matrix

Consider vector spaces V, W with corresponding (ordered) bases $B = (b_1, ..., b_n)$ and $C = (c_1, ..., c_m)$. Moreover, we consider a linear mapping $\Phi : V \longrightarrow W$. For $j \in \{1, ..., n\}$,

$$\Phi(b_j) = \alpha_{1j}c_1 + \cdots + \alpha_{mj}c_m$$

is the unique representation of $\Phi(b_j)$ with respect to C. Then, we call the $m \times n$-matrix A_Φ, whose elements are given by:

$$A_\Phi(i, j) = \alpha_{ij}$$

the transformation matrix of Φ

Matrix Operations

- Basic operations: +, -, *
- Transpose: swap a_{ij} with a_{ji}
- Trace: $tr(A) = \sum a_{ii}$
- Determinant : $det(A) = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma_j}$

Properties

- $A = (A^T)^T$
- $(A + B)^T = A^T + B^T$
- $(\lambda A)^T = \lambda A^T$
- $(A * B)^T = B^T * A^T$

- $tr(A) = tr(A^T)$
- $tr(A + B) = tr(A) + tr(B)$
- $tr(\lambda A) = \lambda tr(A)$
- $tr(A * B) = tr(B * A)$

Properties

- $det(A) = det(A^T)$
- $det(cA) = c^n det(A)$
- $det(A * B) = det(A)det(B)$
- $det(A^k) = (det(A))^k$
- The determinant of a triangular matrix is the product of its diagonal entries.

Special Matrices

- Diagonal

$$D = \begin{bmatrix} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{nn} \end{bmatrix}$$

- Upper and lower triangular
- Symmetric: $a_{ij} = a_{ji}$
- Positive Definite: $x^T Ax > 0$
- Diagonally dominant by rows: $|a_{ii} \geq \sum_{j=1, j \neq j}^{n} |a_{ij}|$
- Orthogonal: $A^T A = AA^T = I$

Norm and Inner Product

A norm on a vector space V is a function:

$$\|.\| : V \longrightarrow R$$

- Positive definite: $\|x\| \geq 0$ and $\|x\| = 0 \iff x = 0$
- Triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$
- Absolutely homogeneous: $\|\lambda x\| = |\lambda| \|x\|$

Manhattan Norm

The Manhattan norm on R^n is defined for $x \in R^n$ as:

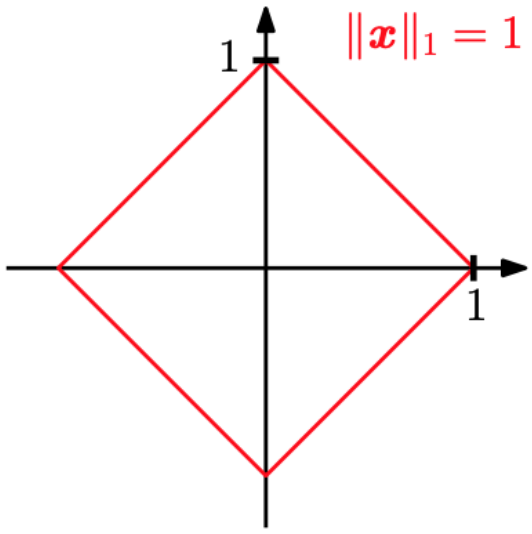
$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

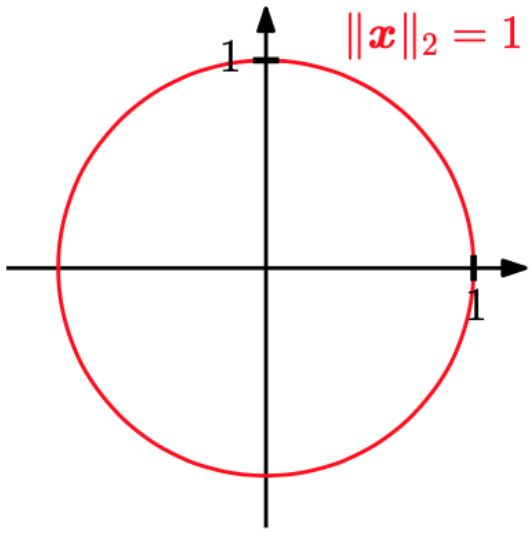
Euclidean Norm

The Euclidean norm on R^n is defined for $x \in R^n$ as:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Manhattan Norm v.s Euclidean Norm





$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$$

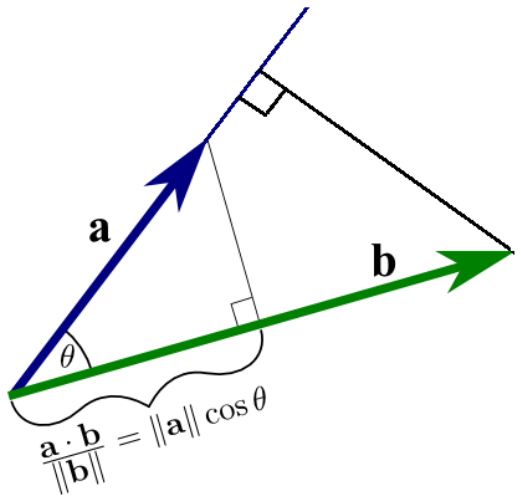
$$\langle x, x \rangle = \sum_{i=1}^{n} x_i^2 = \|x\|_2^2$$

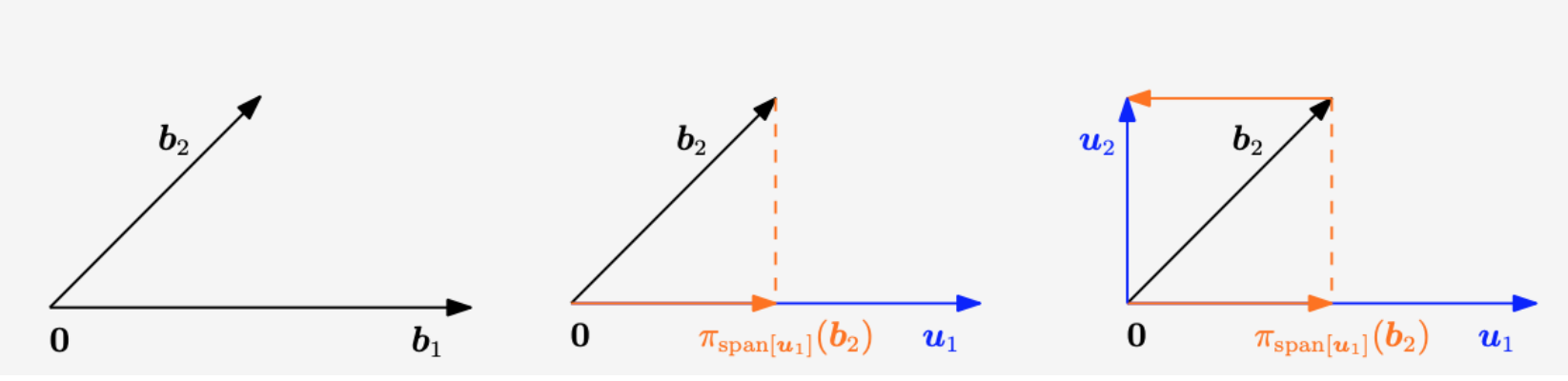
$$\cos\omega = \frac{\langle x, y\rangle}{\|x\|\|y\|}$$

$$x \perp y \longrightarrow \omega = \frac{\pi}{2} \Longrightarrow \cos\frac{\pi}{2} = 0 \longrightarrow \langle x, y\rangle = 0$$

Inner product & projection

$$\pi_b(a) = \|a\| \cos\theta = \frac{\langle a, b \rangle}{\|b\|}$$




b_2
0
b_1
b_2
0
$\pi_{\text{span}[u_1]}(b_2)$
u_1
u_2
b_2
0
$\pi_{\text{span}[u_1]}(b_2)$
u_1

$$u_1 := b_1$$

$$u_k := b_k - \sum_{j=1}^{k-1} \pi_{u_j}(b_k)$$

Statement of the problem

Typical problems of linear algebra

- $Ax = b$
- $min\|Ax - b\|_2$
- $Ax = \lambda x$

Gaussian elimination to upper triangular system

$$
\begin{aligned}
a_{11}x_1 + a_{22}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + \cdots + a_{1n}x_n &= b_1 \\
&\vdots \\
a_{nn}x_n &= b_n
\end{aligned}
$$

Let $a_{ii} \neq 0$, for all i therefore, $x_n = \frac{b_n}{a_{nn}}$ and:

$$x_{ii} = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} \quad for \quad i = n-1, n-2, \cdots, 1$$

Operation counts: $O(n^2)$

$$a_{11}x_1 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

$$\vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

Let $a_{ii} \neq 0$, for all i therefore, $x_1 = \frac{b_1}{a_{11}}$ and:

$$x_{ii} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}} \quad for \quad i = 2, 3, \cdots, n$$

Operation counts: $O(n^2)$

LU Decomposition

LU Decomposition

The non-singular matrix A has an LU-factorization if it can be expressed as the product of a lower-triangular matrix L and an upper triangular matrix U:

$$A = LU$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{nn} \end{bmatrix}$$

- If $l_{ii} = 1$ for $i = 1, \cdots, n$, then LU-factorization is called Doolittle factorization
- If $u_{ii} = 1$ for $i = 1, \cdots, n$, then LU-factorization is called Crout factorization

Doolittle Factorization

Doolittle Factorization

Algorithm:

For $i = 1, ..., n$ do

For $j = i, ..., n$ do

$$u_{ij} = a_{ij} - \sum_{k=1}^{i=1} l_{ik} u_{kj}$$

For $j = i + 1, ..., n$ do

$$l_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki}}{u_{ii}}$$

Crout Factorization

Algorithm:

For $i = 1, ..., n$ do

For $j = i, ..., n$ do

$l_{ji} = a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki}$

For $j = i + 1, ..., n$ do

$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}}{l_{ii}}$

Decomposition methods to $\mathbf{Ax} = \mathbf{b}$

$$\mathbf{A} = \mathbf{LU}, \quad \mathbf{Ax} = \mathbf{b}, \quad \rightarrow \quad (\mathbf{LU})\mathbf{x} = \mathbf{b},$$

$$\mathbf{L}(\mathbf{Ux}) = \mathbf{b}, \quad \rightarrow \quad \mathbf{Ly} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{y},$$

$$\det(\mathbf{A}) = \det(\mathbf{L}) \det(\mathbf{U})$$

Cholesky Decomposition

Cholesky Decomposition

A symmetric, positive definite matrix A can be factorized into a product $A = LL^T$, where L is a lower triangular matrix with positive diagonal elements:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \cdots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_{nn} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = LL^T = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$A = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & l_{31}l_{21} + l_{32}l_{22} \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix}$$

$$l_{11} = \sqrt{a_{11}}$$

$$l_{21} = \frac{1}{l_{11}} a_{12} \longrightarrow l_{22} = \sqrt{a_{22} - l_{21}^2}$$

$$l_{31} = \frac{1}{l_{11}} a_{13} \longrightarrow l_{32} = \frac{1}{l_{22}} (a_{23} - l_{31} l_{21}) \longrightarrow l_{33} = \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)}$$

QR Decomposition

The QR decomposition of a matrix is a decomposition of the matrix into an orthogonal matrix (Q) and an upper triangular matrix (R). A QR decomposition of a real square matrix A with full rank is a decomposition of A as:

$$A = QR$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{nn} \end{bmatrix}$$

QR factorization

The $n \times n$ matrix $\mathbf{A}^T\mathbf{A}$ is symmetric and positive definite and thus it can be written uniquely as $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ where $\mathbf{L}$ is lower triangular with positive diagonal entries.
We can show that $\mathbf{Q} = \mathbf{A}(\mathbf{L}^T)^{-1}$ is an orthogonal matrix. Then $\mathbf{A} = \mathbf{Q}\mathbf{L}^T$ so set $\mathbf{R} = \mathbf{L}^T$ and we are done because $\mathbf{L}$ has positive diagonal entries.

QR factorization

The $\mathbf{QR}$ decomposition can be used to solve a linear system $\mathbf{Ax} = \mathbf{b}$. We have

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{QRx} = \mathbf{b}, \quad \mathbf{Q}^T\mathbf{QRx} = \mathbf{Q}^T\mathbf{b},$$

then

$$\mathbf{Rx} = \mathbf{Q}^T\mathbf{b},$$

which is an upper triangular matrix.

What is Eigenvalue?

Let $A \in R^{n \times n}$ be a square matrix. Then $\lambda \in R$ is an eigenvalue of A and $x \in R^n - \{0\}$ is the corresponding eigenvector of A if:

$$Ax = \lambda x$$

Characteristic polynomial

$$p(A) = det(A - \lambda I) = c_0 + c_1\lambda + c_2\lambda^2 + \cdot + c_n\lambda^{n-1} + (-1)^n\lambda^n$$

$$Ax = \lambda x$$

$$Ax - \lambda x = 0$$

$$(A - \lambda I)x = 0$$

Characteristic polynomial and eigenvalue

$$Ax = \lambda x$$

$$Ax - \lambda x = 0$$

$$(A - \lambda I)x = 0$$

$$det(A - \lambda I) = ?$$

Characteristic polynomial and eigenvalue

$$Ax = \lambda x$$

$$Ax - \lambda x = 0$$

$$(A - \lambda I)x = 0$$

$$det(A - \lambda I) = 0$$

Remark

The homogeneous system $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ has a nontrivial solution iff

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

$P_A(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$ is a polynomial in λ of degree n and is called the characteristic polynomial of $\mathbf{A}$. Thus the n eigenvalues of $\mathbf{A}$ are the n roots of its characteristic polynomial.

Eigenvalue Decomposition

Similar matrices

Two matrices A and D are similar if there exists an invertible matrix P, such that:

$$D = P^{-1}AP$$

.

D is a diagonal matrix

$$D = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

$$P = [p_1, p_2, \cdots, p_n]$$

$$D = P^{-1}AP$$

$$PD = AP$$

D is a diagonal matrix

$$AP = A[p_1, p_2, \cdots, p_n] = [Ap_1, Ap_2, \cdots, Ap_n]$$

$$PD = [p_1, p_2, \cdots, p_n] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} = [\lambda_1 p_1, \lambda_2 p_2, \cdots, \lambda_n p_n]$$

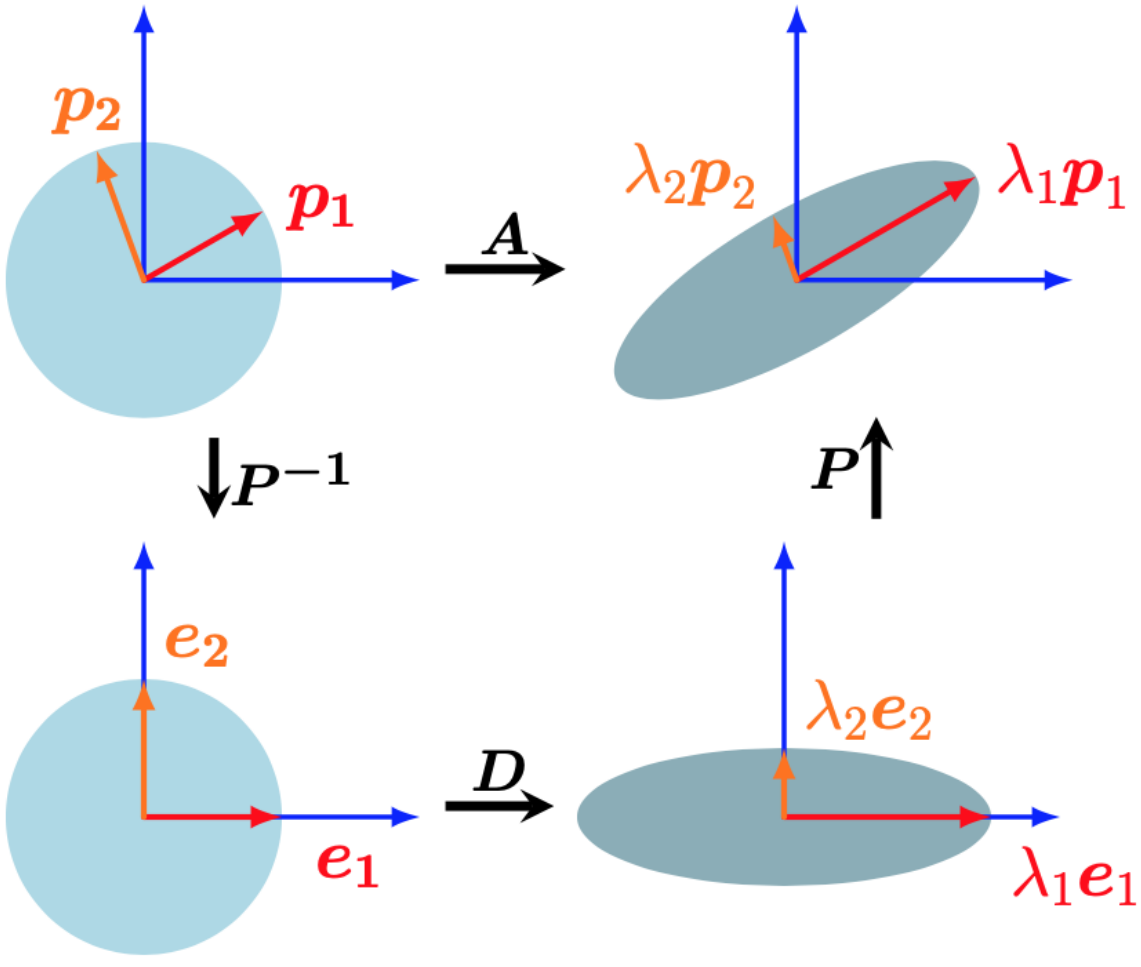
$$Ap_1 = \lambda_1 p_1$$

$$Ap_2 = \lambda_1 p_2$$

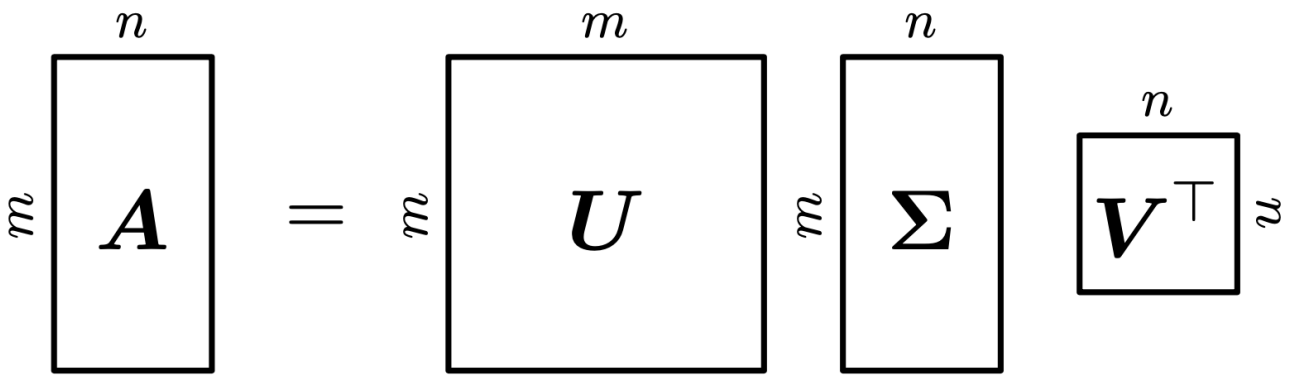
$$\vdots$$

$$Ap_n = \lambda_1 p_n$$

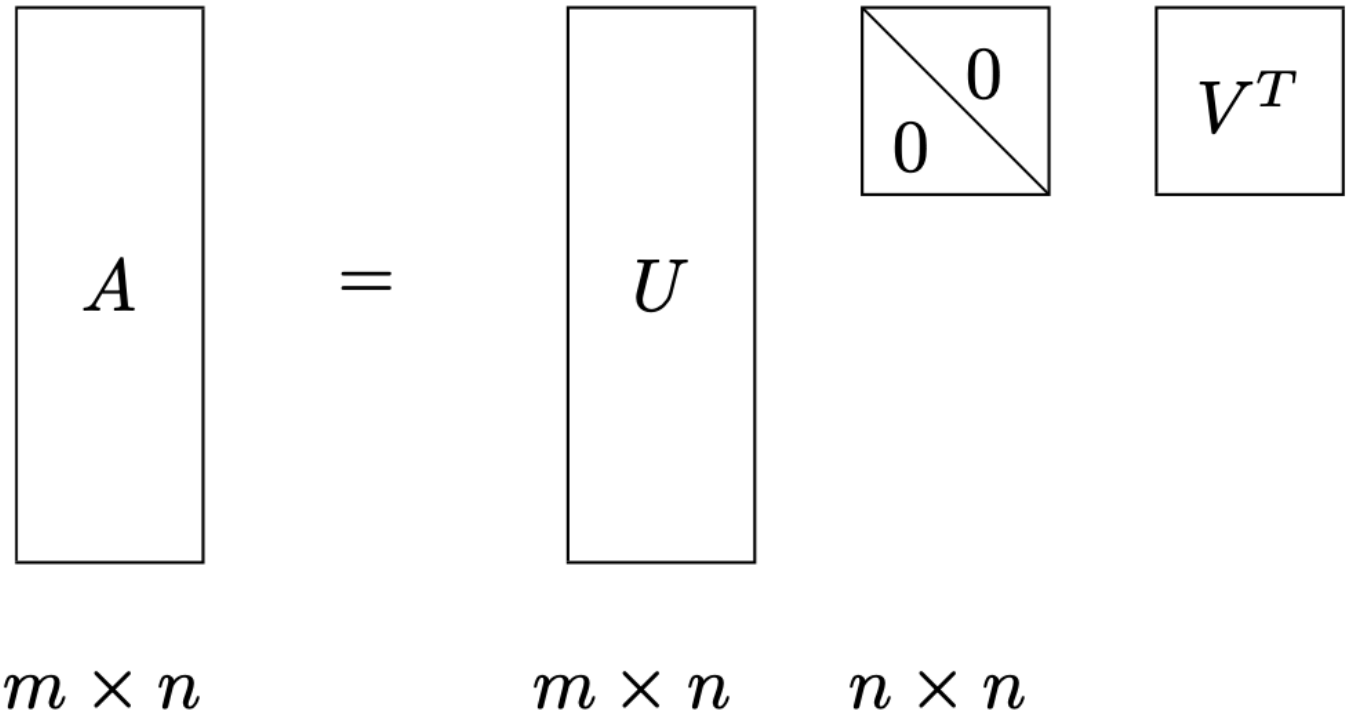
Eigenvalue Decomposition



Singular Value Decomposition


n
m
A
=
m
U
m
n
Σ
n
n
V^⊤

One rank approximation



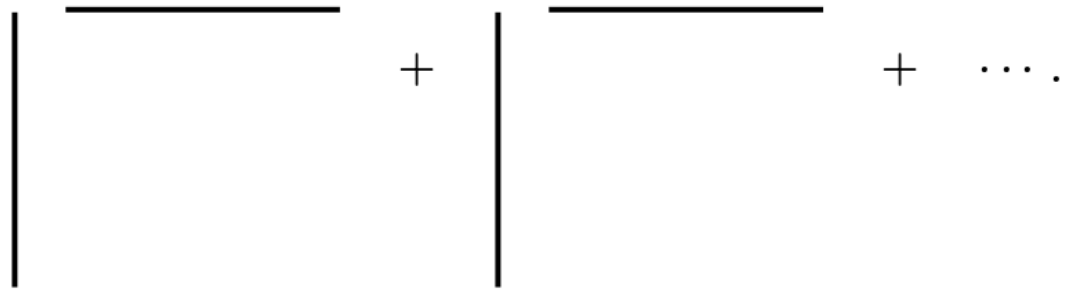
One rank approximation

$$A = U_1 \Sigma V^T = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix} \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n^T \end{pmatrix} \begin{pmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{pmatrix}$$

$$= \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix} \begin{pmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \\ \vdots \\ \sigma_n v_n^T \end{pmatrix} = \sum_{i=1}^{n} \sigma_i u_i v_i^T .$$

One rank approximation

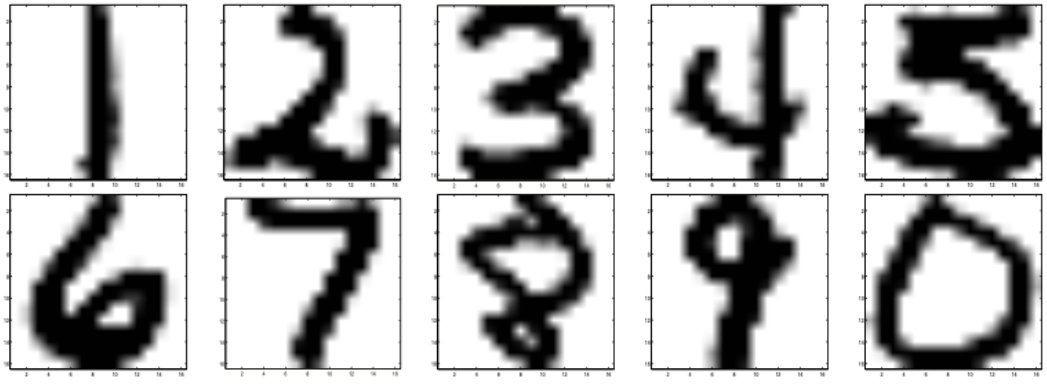
$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T =$$



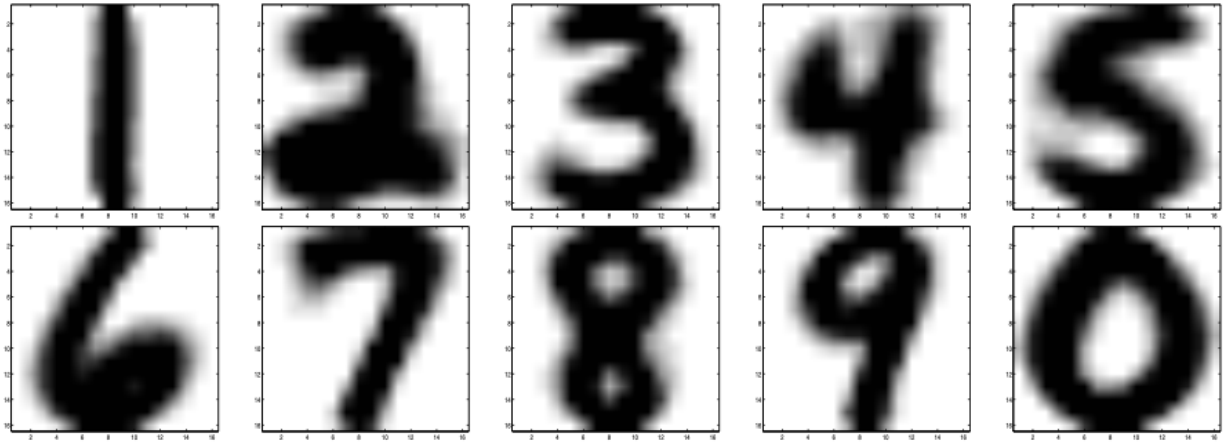
$$+ \quad + \quad \cdots .$$

$$AA^T = U\Sigma V^T (U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$$
$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T$$

Classification using SVD



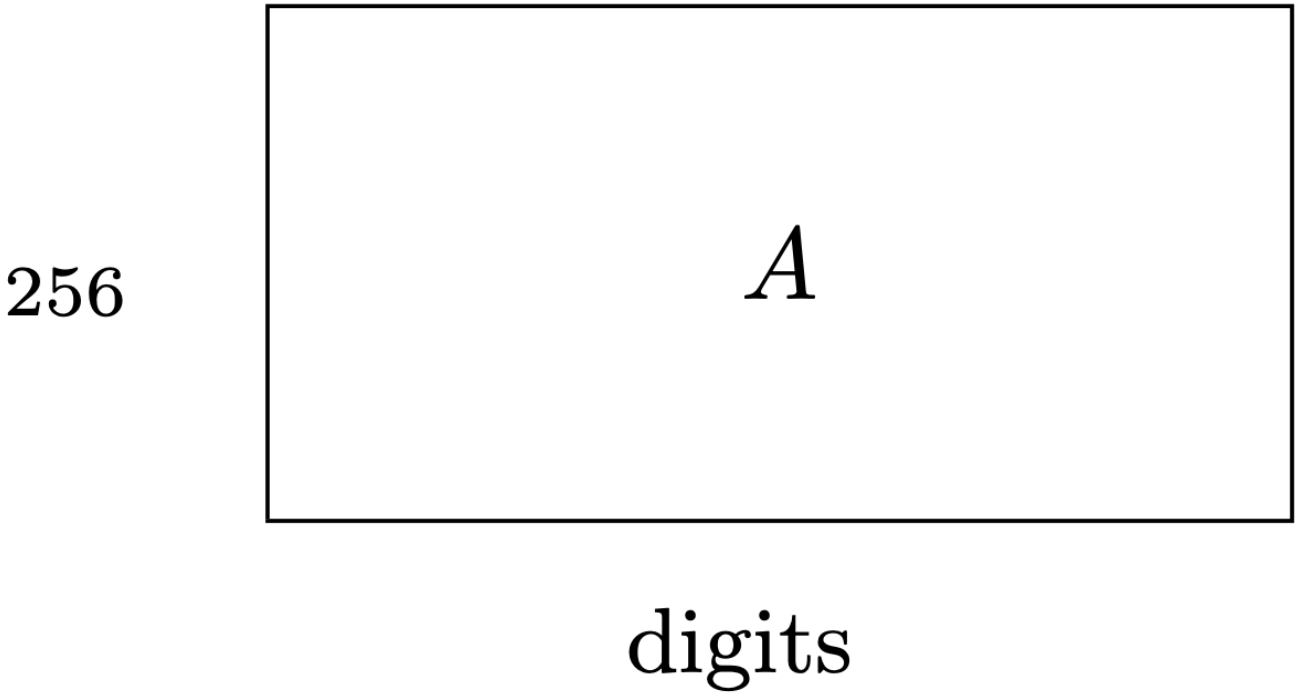
Classification using SVD



A simple classification algorithm

Training: Given the manually classified training set, compute the means (centroids) m_i, $i = 0, \ldots, 9$, of all the 10 classes.

Classification: For each digit in the test set, classify it as k if m_k is the closest mean.



$$\min_{\alpha_i} \left\| z - \sum_{i=1}^{k} \alpha_i u_i \right\|,$$

$$\min_{\alpha} \| z - U_k \alpha \|_2,$$

$$\| (I - U_k U_k^T) z \|_2,$$

An SVD basis classification algorithm

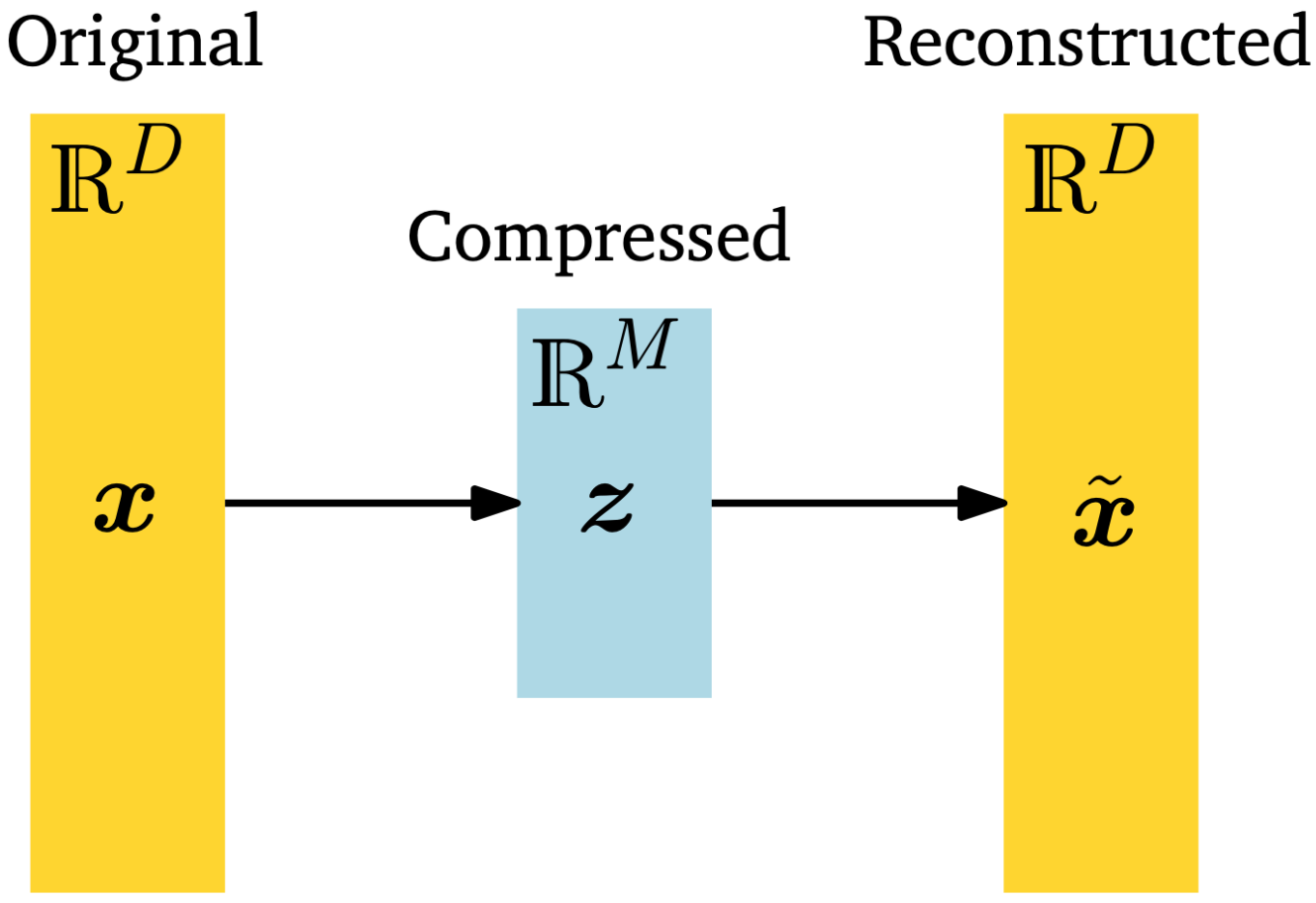
Training: For the training set of known digits, compute the SVD of each set of digits of one kind.

Classification: For a given test digit, compute its relative residual in all 10 bases. If one residual is significantly smaller than all the others, classify as that. Otherwise give up.

Principal Component Analysis

Why is dimension reduction important?

- The goal for any dimensional reduction method is to reduce the dimensions of the original data for different purposes such as visualization, decrease CPU time, ..etc..
- Dimensionality reduction techniques are important in many applications related to machine learning, data mining, Bioinformatics, biometric and information retrieval.


Original
Reconstructed
$\mathbb{R}^D$
Compressed
$\mathbb{R}^M$
$\mathbb{R}^D$
$\boldsymbol{x}$
$\boldsymbol{z}$
$\tilde{\boldsymbol{x}}$

Consider an i.i.d. dataset $X = \{x_1, ..., x_N\}$, $x_n \in R^D$, with mean 0 that possesses the data covariance matrix:

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n x_n^T$$

Furthermore, we assume there exists a low-dimensional compressed representation:

$$z_n = W^T x_n \in R^M$$

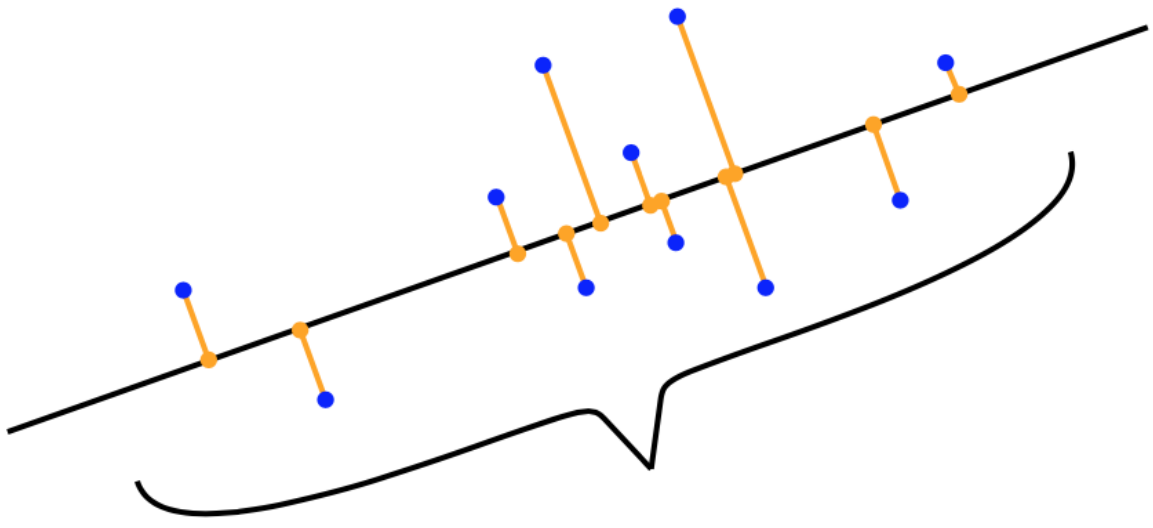
The projection matrix is defined as follows:

$$W = [w_1, \cdots, w_M] \in R^{D \times M}$$

- $w_i^T w_j = 0$, if and only if $j \neq i$
- $w_i^T w_i = 1$

Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code (Hotelling, 1933).

Maximum Variance Perspective



PCA finds a lower-dimensional subspace (line) that maintains as much variance (spread of the data) as possible when the data (blue) is projected onto this subspace (orange).

We maximize the variance of the low-dimensional code using a sequential approach. We start by seeking a single vector $w_1 \in R^D$ that maximizes the variance of the projected data, i.e., we aim to maximize the variance of the first coordinate z_1 of $z \in R^M$ so that

$$V_1 = V[z_1] = \frac{1}{N} \sum_{n=1}^{N} z_{1n}^2$$

$$z_{1n} = w_1^T x_n$$

$$V_1 = V[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2 = \frac{1}{N}\sum_{n=1}^{N}(w_1^T x_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(w_1^T x_n x_n^T w_1)$$

$$= w_1^T \left(\frac{1}{N}\sum_{n=1}^{N} x_n x_n^T\right) w_1 = w_1^T S w_1$$

$$\max_{w_1} \; w_1^T S w_1$$

$$subject\ to\ \|w_1\|^2 = 1$$

$$L(w_1, \lambda_1) = w_1^T S w_1 + \lambda_1(1 - w_1^T w_1)$$

$$\frac{\partial L}{\partial w_1} = 2w_1^T S - 2\lambda_1 w_1^T$$

$$\frac{\partial L}{\partial \lambda_1} = 1 - w_1^T w_1$$

$$\frac{\partial L}{\partial w_1} = 0 \longrightarrow Sw_1 = \lambda_1 w_1$$

$$\frac{\partial L}{\partial \lambda_1} = 0 \longrightarrow w_1^T w_1 = 1$$

$$Sw_1 = \lambda_1 w_1$$

$$w_1^T w_1 = 1$$

$$V_1 = w_1^T S w_1 = \lambda_1 w_1^T w_1 = \lambda_1$$

We can determine the effect/contribution of the principal component w_1 in the original data space by mapping the coordinate z_{1n} back into data space, which gives us the projected data point

$$\tilde{x}_n = w_1 z_{1n} = w_1 w_1^T x_n \in R^D$$

PCA in Action

- Step 1: Get some data
- Step 2: Subtract the mean
- Step 3: Calculate the covariance matrix
- Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix
- Step 5: Choosing components and forming a feature vector
- Step 6: Deriving the new data set

PCA in Action: Step 1

$$data = \begin{bmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2 & 1.6 \\ 1 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{bmatrix}$$

PCA in Action: Step 2

$$dataAdjust = \begin{bmatrix} .69 & .49 \\ -1.31 & -1.21 \\ .39 & .99 \\ .09 & .29 \\ 1.29 & 1.09 \\ .49 & .79 \\ .19 & -.31 \\ -.81 & -.81 \\ -.31 & -.31 \\ -.71 & -1.01 \end{bmatrix}$$

PCA in Action: Step 3

$$cov = \begin{bmatrix} .6165 & .6154 \\ .6154 & .7165 \end{bmatrix}$$

$$eigenvalues = \begin{bmatrix} 1.2840 \\ .0490 \end{bmatrix}$$

$$eigenvectors = \begin{bmatrix} -.6778 & -.7351 \\ -.7351 & .6778 \end{bmatrix}$$

PCA in Action: Step 5

$$TransformedData = dataAdjust \times eigenvector$$

$$= \begin{bmatrix} .69 & .49 \\ -1.31 & -1.21 \\ .39 & .99 \\ .09 & .29 \\ 1.29 & 1.09 \\ .49 & .79 \\ .19 & -.31 \\ -.81 & -.81 \\ -.31 & -.31 \\ -.71 & -1.01 \end{bmatrix} \begin{bmatrix} -.6778 \\ -.7351 \end{bmatrix} = \begin{bmatrix} -.827970186 \\ 1.77758033 \\ -.992197494 \\ -.274210416 \\ -1.67580142 \\ -.912949103 \\ .0991094375 \\ 1.14457216 \\ .438046137 \\ 1.22382056 \end{bmatrix}$$

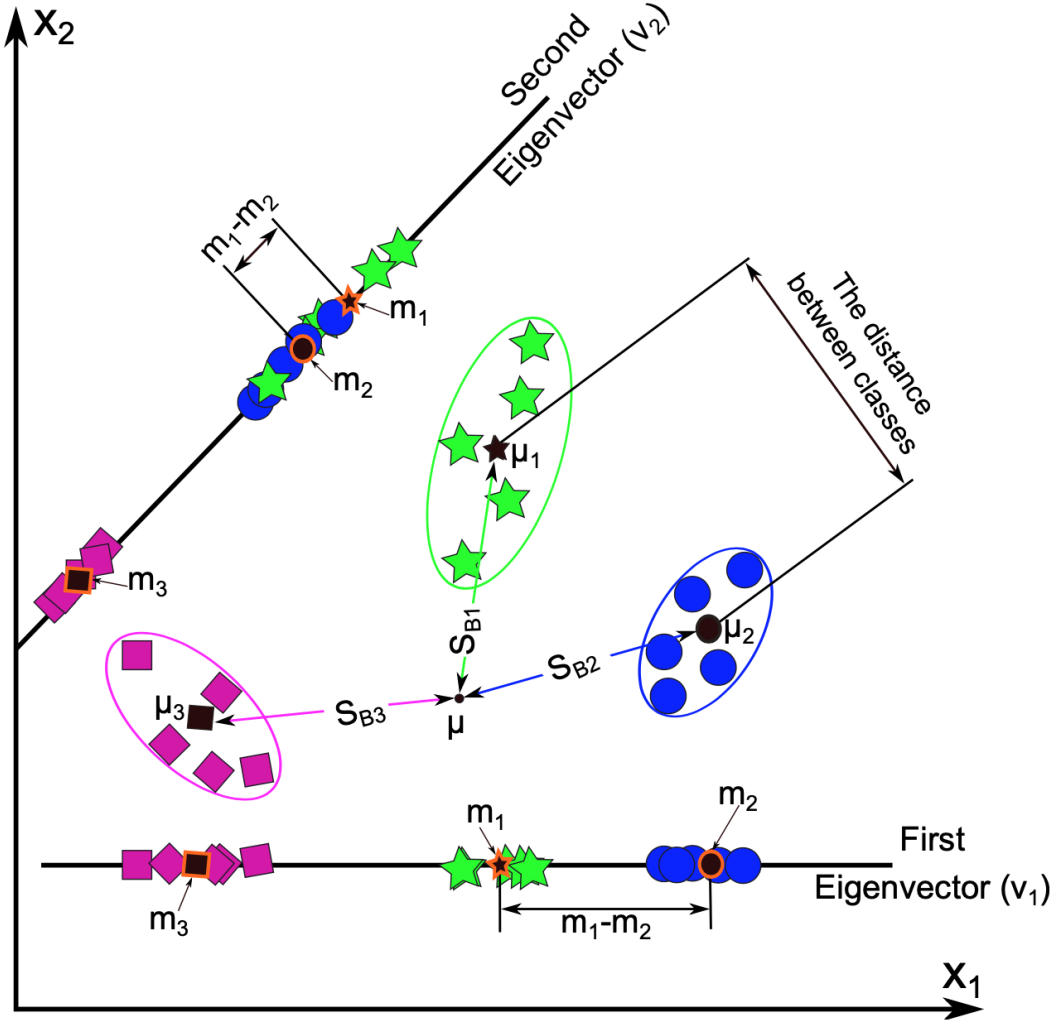
Linear Discriminant Analysis

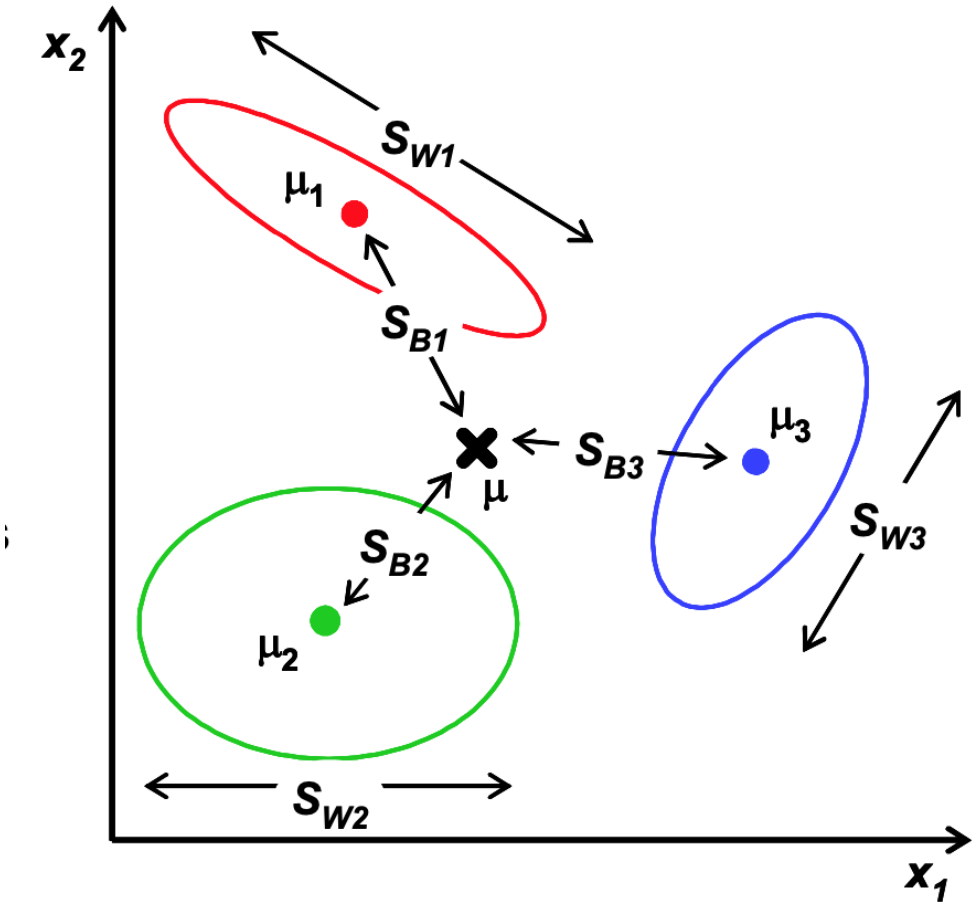
$$X = \{x_1, x_2, \cdots, x_N\} \longrightarrow X = [w_1, w_2, \cdots, w_c]$$

Goal of LDA

The Linear Discriminant Analysis (LDA) technique is developed to transform the features into a lower dimensional space, which maximizes the ratio of the between-class variance to the within-class variance, thereby guaranteeing maximum class separability.

$$J(W) = \frac{W^T S_B W}{W^T S_W W}$$


X2
Second
Eigenvector (v2)
m1-m2
m1
m2
The distance
between classes
μ1
m3
SB1
SB2
μ2
μ3
SB3
μ
m1
m2
First
Eigenvector (v1)
m3
m1-m2
X1


x_2
S_{W1}
μ_1
S_{B1}
S_{B3}
μ_3
μ
S_{W3}
S_{B2}
μ_2
S_{W2}
x_1

Maximizing the objective

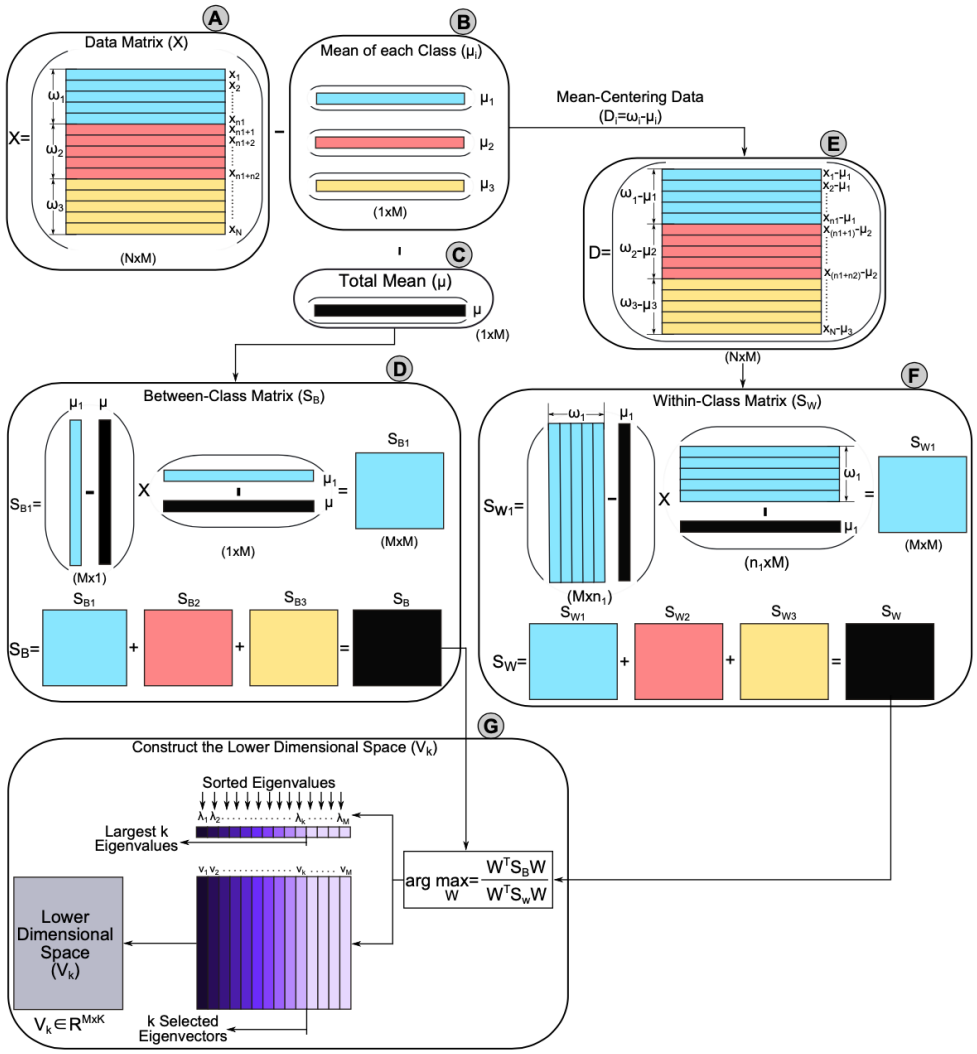
$$J(W) = \frac{W^T S_B W}{W^T S_W W}$$

$$\frac{dJ}{dW} = 0 \Longleftrightarrow (W^T S_W W) S_B W - (W^T S_B W) S_W W = 0$$

$$S_B W - J S_W W = 0 \Longrightarrow S_W^{-1} S_B W = JW$$

- The projection vector w_i is the eigenvector of $S_W^{-1} S_B$

LDA steps



LDA steps

- Mean normalization
- Compute mean vectors classes
- Compute scatter matrices S_W, S_B
- Compute eigenvectors and eigenvalues of $S_W^{-1} S_B$
- Select k eigenvectors with the largest eigenvalues to form transform matrix
- Project samples onto the new subspace using W and compute the new coordinates

$$\omega_1 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 3 \\ 4 & 5 \\ 5 & 5 \end{bmatrix}, \quad \omega_2 = \begin{bmatrix} 4 & 2 \\ 5 & 0 \\ 5 & 2 \\ 3 & 2 \\ 5 & 3 \\ 6 & 3 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 3 & 3.6 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 4.67 & 2 \end{bmatrix}$$

$$\mu = [\frac{5}{11}\mu_1 \quad \frac{6}{11}\mu_2] = [3.91 \quad 2.727]$$

$$S_{B_1} = n_1(\mu_1 - \mu)^T(\mu_1 - \mu) = 5[-0.91 \quad 0.87]^T[-0.91 \quad 0.87]$$

$$= \begin{bmatrix} 4.13 & -3.97 \\ -3.97 & 3.81 \end{bmatrix}$$

Similarly,

$$S_{B_2} = \begin{bmatrix} 3.44 & -3.31 \\ -3.31 & 3.17 \end{bmatrix}$$

$$S_B = S_{B_1} + S_{B_2} = \begin{bmatrix} 4.13 & -3.97 \\ -3.97 & 3.81 \end{bmatrix} + \begin{bmatrix} 3.44 & -3.31 \\ -3.31 & 3.17 \end{bmatrix}$$

$$= \begin{bmatrix} 7.58 & -7.27 \\ -7.27 & 6.98 \end{bmatrix}$$

$$d_1 = \begin{bmatrix} -2. & -1.6 \\ -1 & -0.6 \\ 0 & -0.6 \\ 1 & 1.4 \\ 2 & 1.4 \end{bmatrix}, \quad d_2 = \begin{bmatrix} -0.67 & 0 \\ 0.33 & -2 \\ 0.33 & 0 \\ -1.67 & 0 \\ 0.33 & 1 \\ 1.33 & 1 \end{bmatrix}$$

$$S_{W_j} = d_j^T d_i, \qquad S_W = \sum_{j=1}^{c} S_{W_j}$$

$$S_{W_1} = \begin{bmatrix} 10 & 8 \\ 8 & 7.2 \end{bmatrix}, \quad S_{W_2} = \begin{bmatrix} 5.3 & 1 \\ 1 & 6 \end{bmatrix}$$

$$S_W = \begin{bmatrix} 15.3 & 9 \\ 9 & 13.2 \end{bmatrix}$$

$$S_W^{-1} = \begin{bmatrix} 0.11 & -0.07 \\ -0.07 & 0.13 \end{bmatrix}, \quad S_B = \begin{bmatrix} 7.58 & -7.27 \\ -7.27 & 6.98 \end{bmatrix}$$

$$S_W^{-1} S_B = \begin{bmatrix} 1.37 & -1.32 \\ -1.49 & 1.43 \end{bmatrix} \longrightarrow \lambda_1 = 2.81, \lambda_2 = -0.0027$$

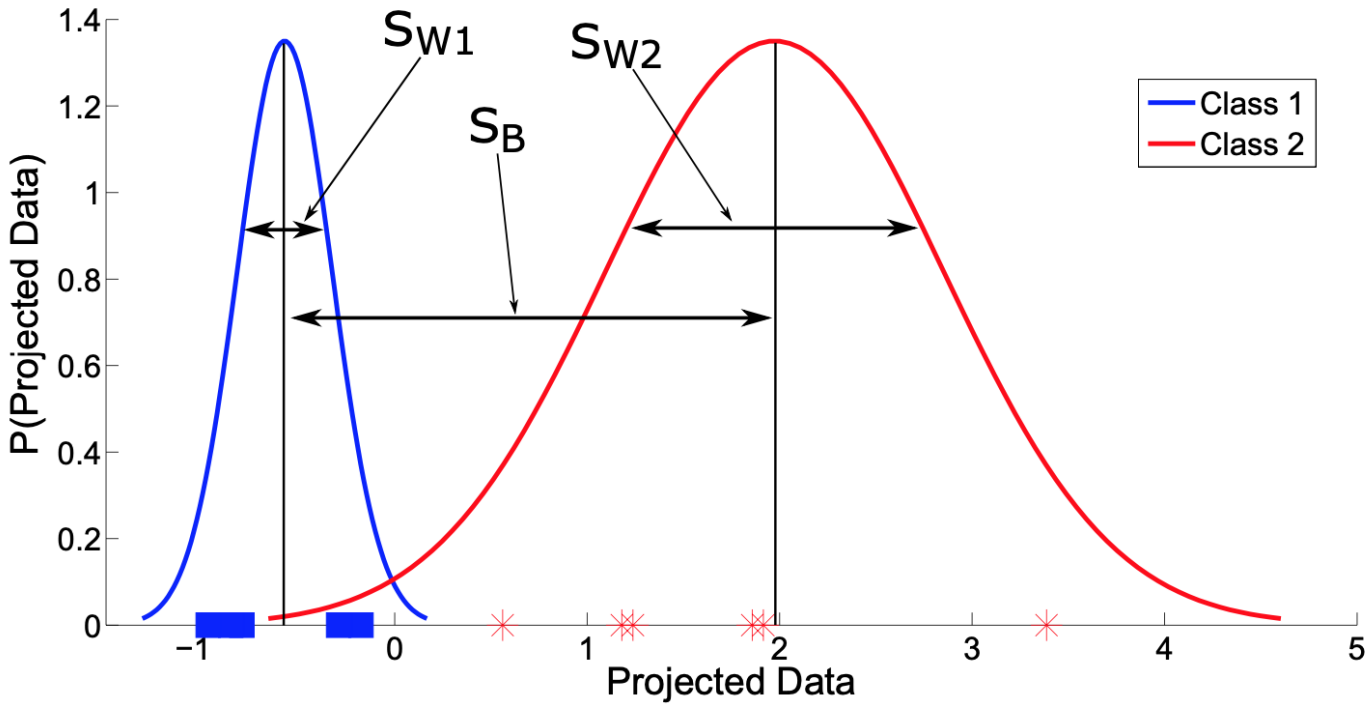
$$W = \begin{bmatrix} 0.68 & -0.69 \\ -0.74 & -0.72 \end{bmatrix}$$

$$y_i = \omega_i W_1$$

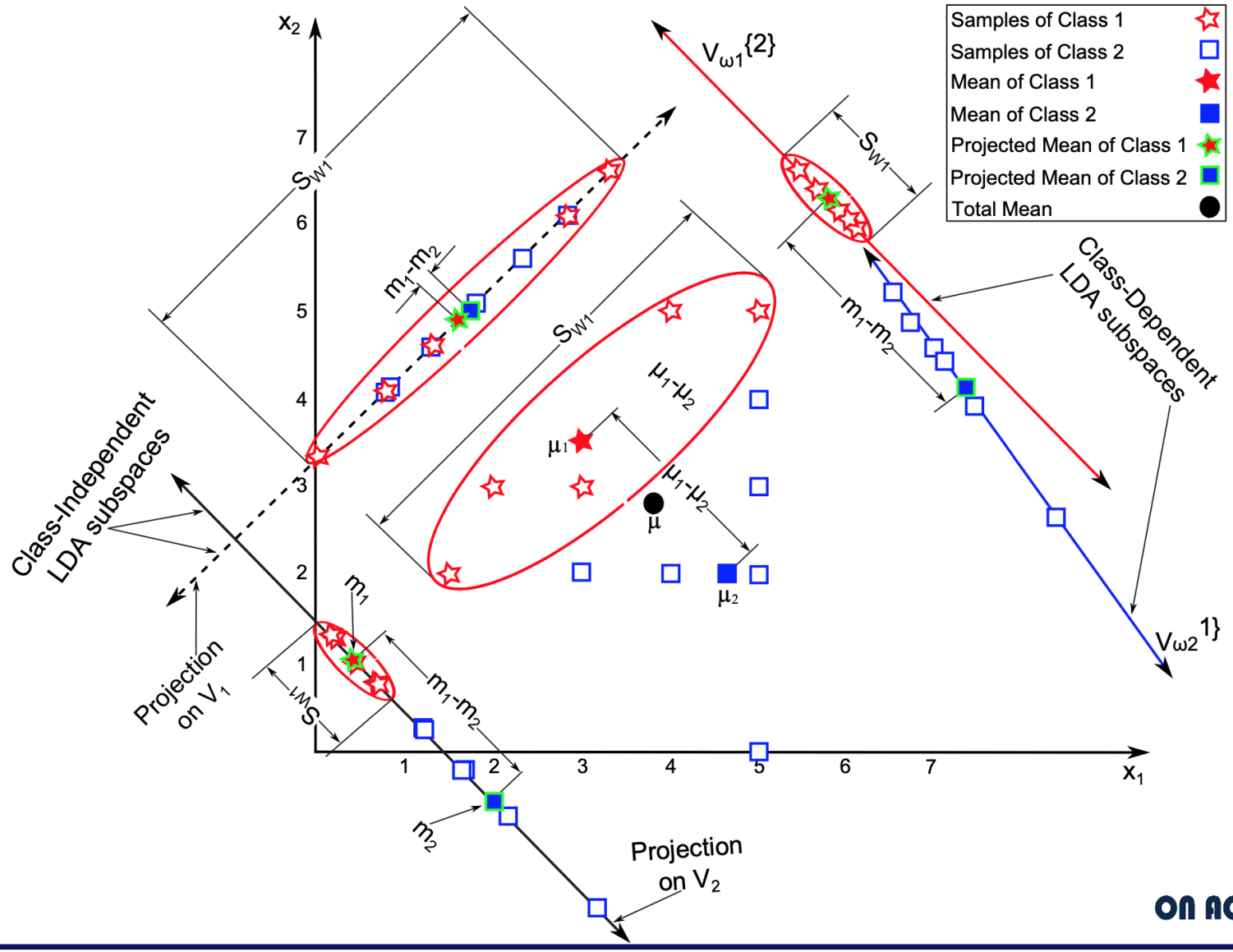
$$y_1 = \omega_1 W_1 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 3 \\ 4 & 5 \\ 5 & 5 \end{bmatrix} \begin{bmatrix} 0.68 \\ -0.74 \end{bmatrix} = \begin{bmatrix} -0.79 \\ -0.85 \\ -0.18 \\ -0.97 \\ -0.29 \end{bmatrix}$$

$$y_2 = \omega_2 W_1 = \begin{bmatrix} 1.24 \\ 3.39 \\ 1.92 \\ 0.56 \\ 1.18 \\ 1.86 \end{bmatrix}$$

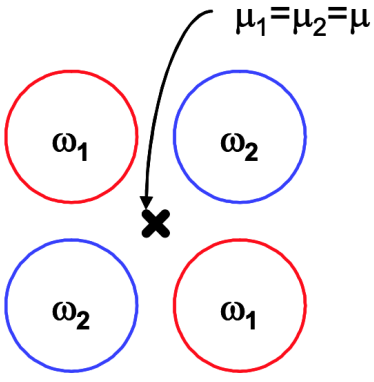
LDA in Action

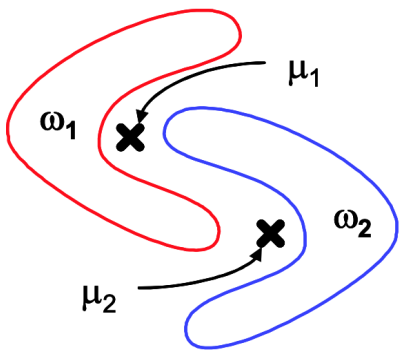


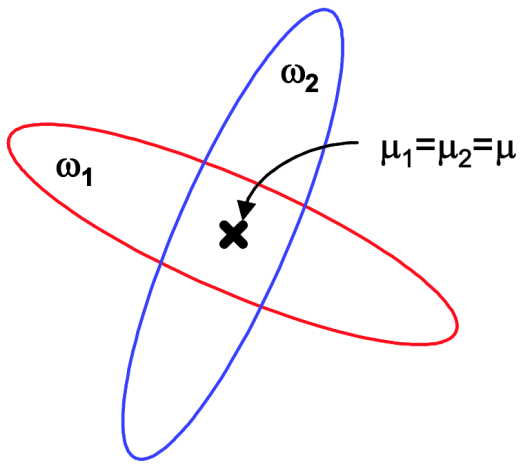
LDA in Action



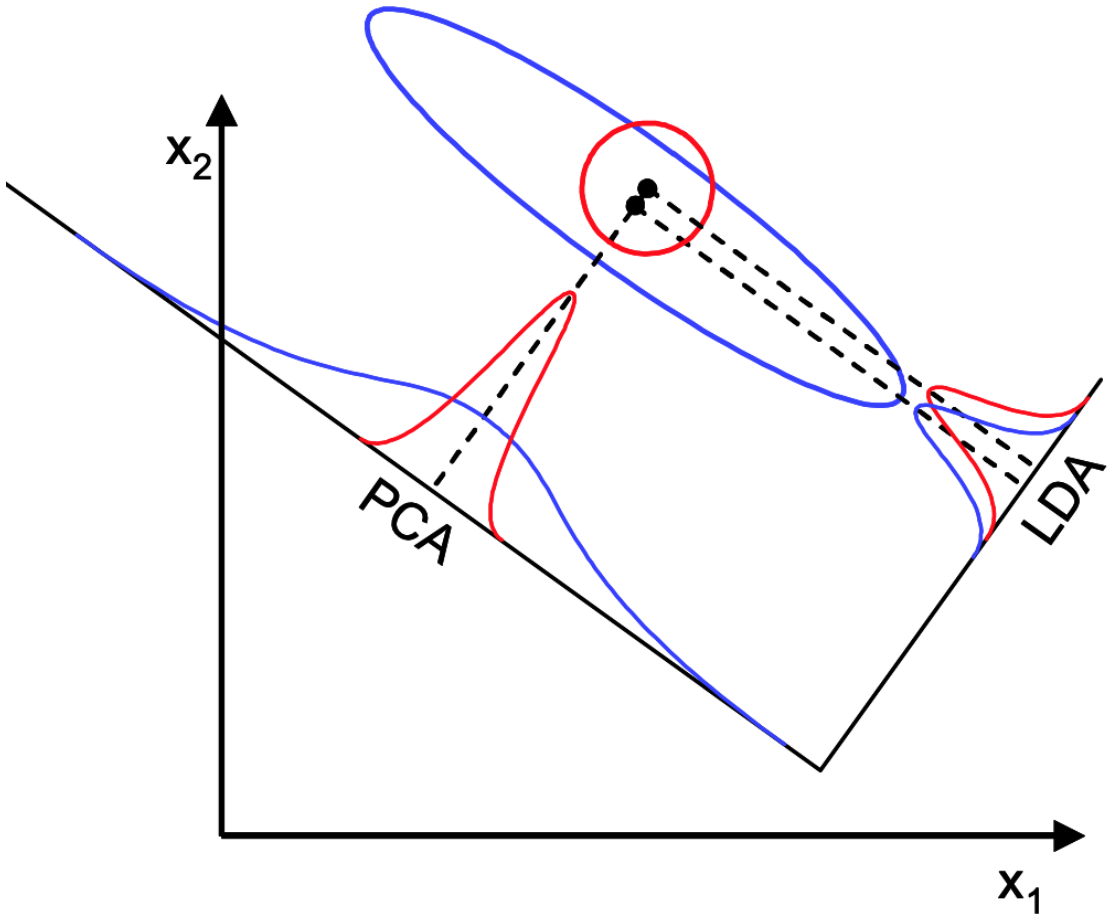
Limitations of LDA



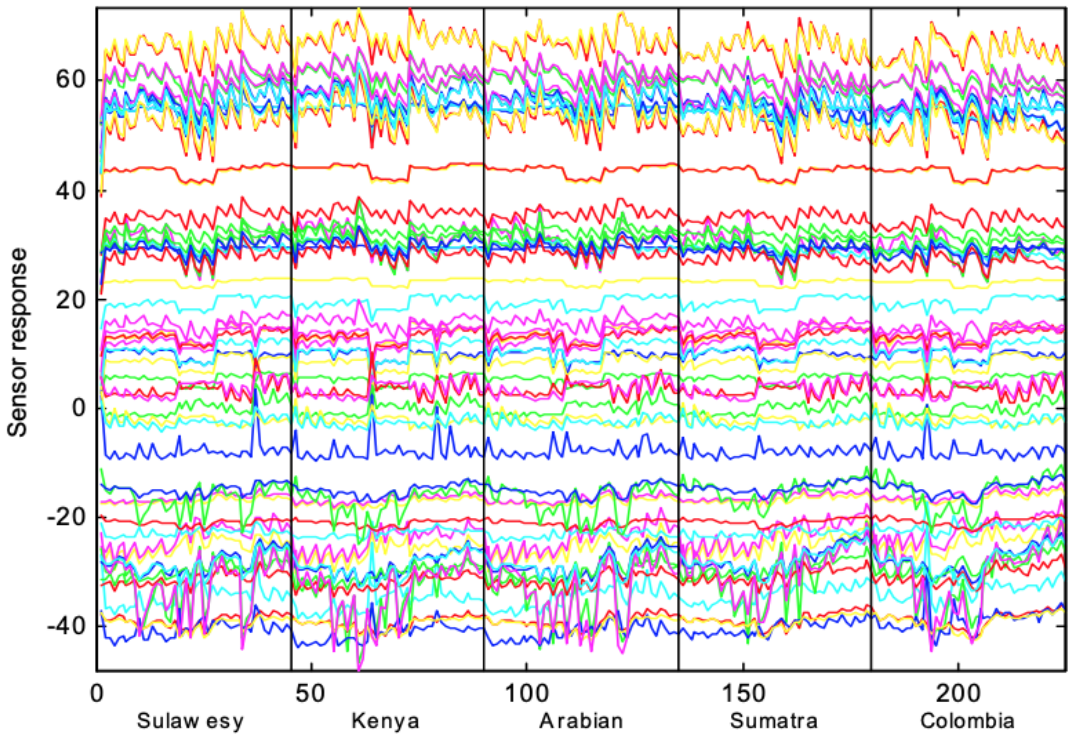


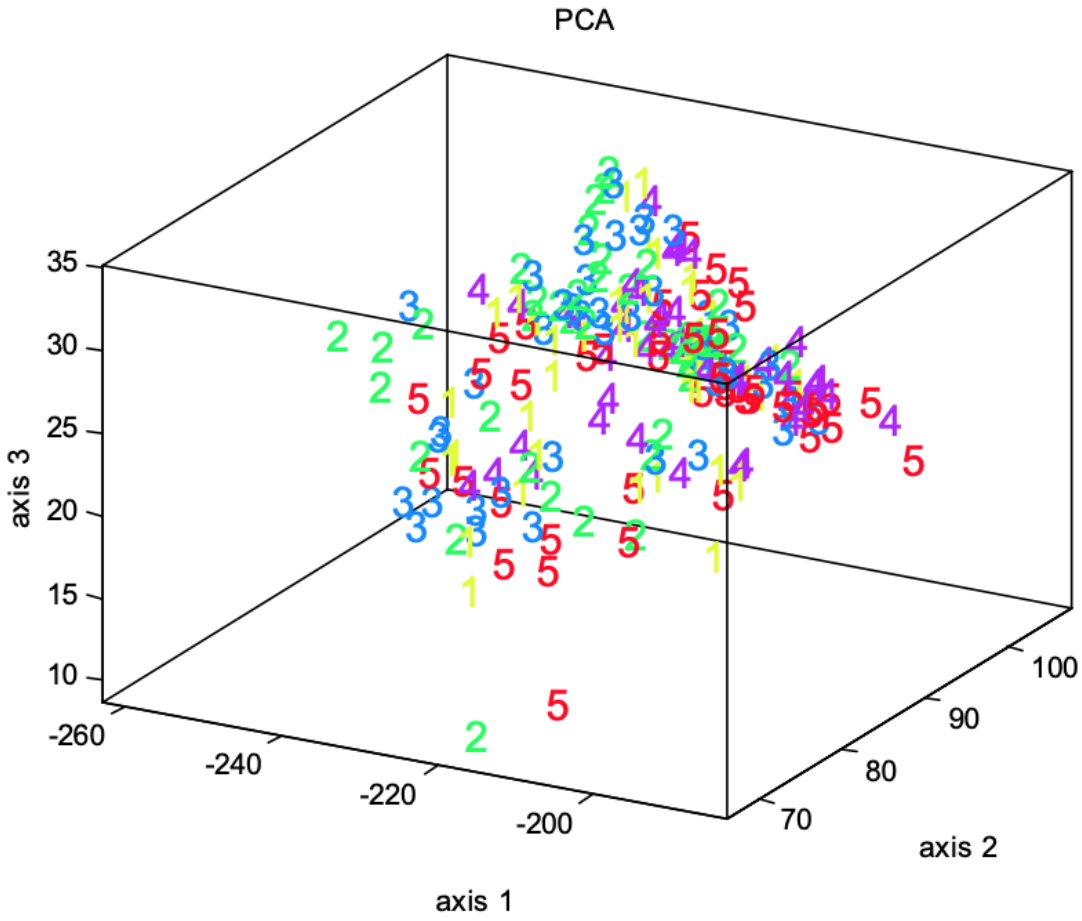


Limitations of LDA

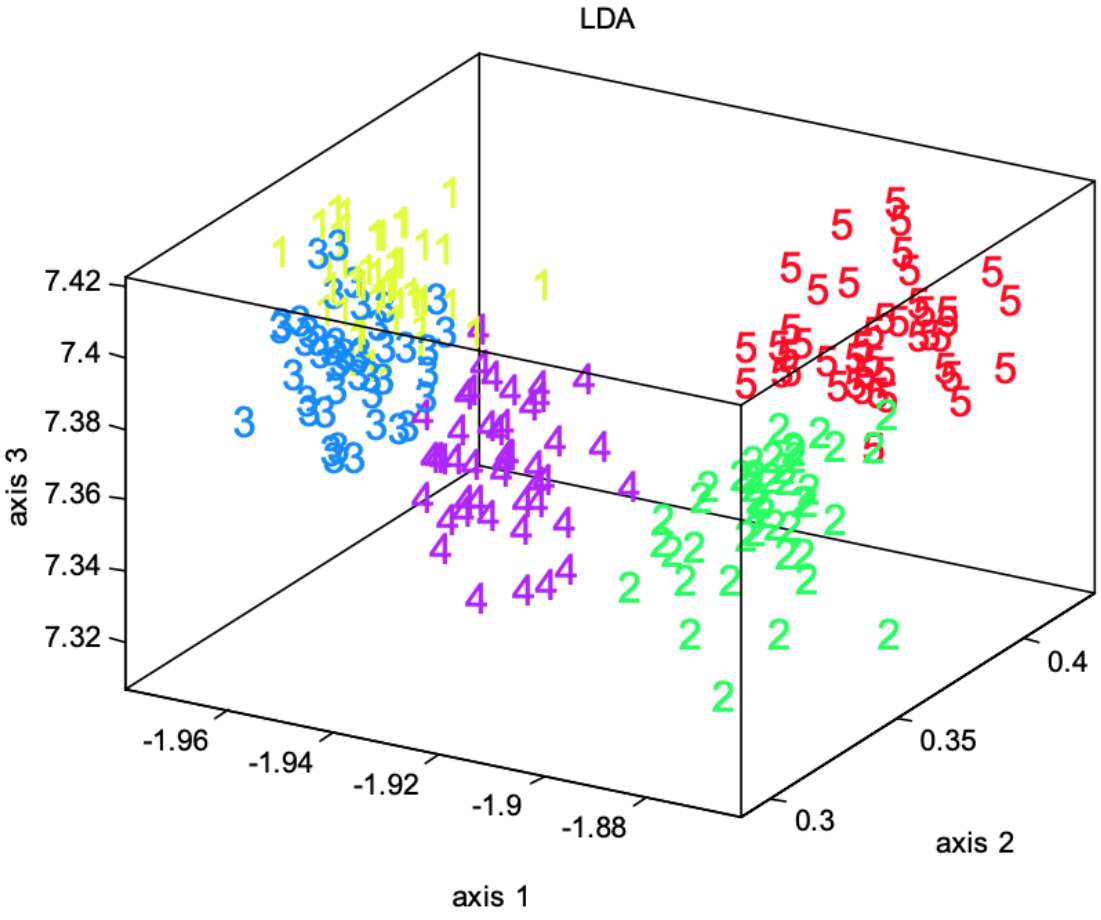


LDA Vs. PCA

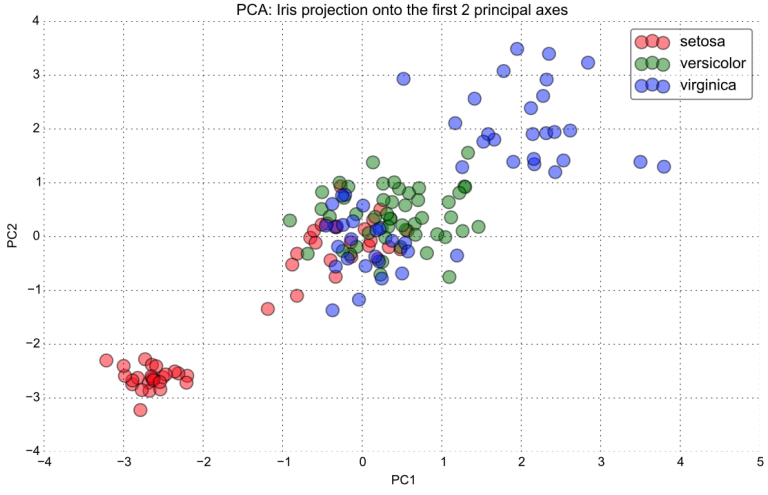


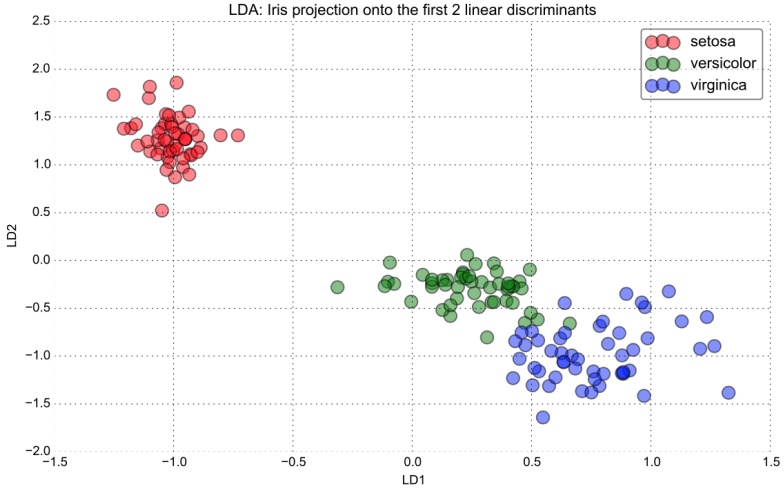

PCA
axis 3
35
30
25
20
15
10
-260
-240
-220
-200
axis 1
70
80
90
100
axis 2

LDA Vs. PCA


LDA
7.42
7.4
7.38
7.36
7.34
7.32
axis 3
-1.96
-1.94
-1.92
-1.9
-1.88
axis 1
0.4
0.35
0.3
axis 2

LDA Vs. PCA


PCA: Iris projection onto the first 2 principal axes
setosa
versicolor
virginica
PC1
PC2


LDA: Iris projection onto the first 2 linear discriminants
setosa
versicolor
virginica
LD1
LD2

LDA Vs. PCA

- LDA: Magnitude of the eigenvalues in LDA describe importance of the corresponding eigenspace with respect to classification performance
- PCA: Magnitude of the eigenvalues in PCA describe importance of the corresponding eigenspace with respect to minimizing reconstruction error

Thank you for your attentions! ☺