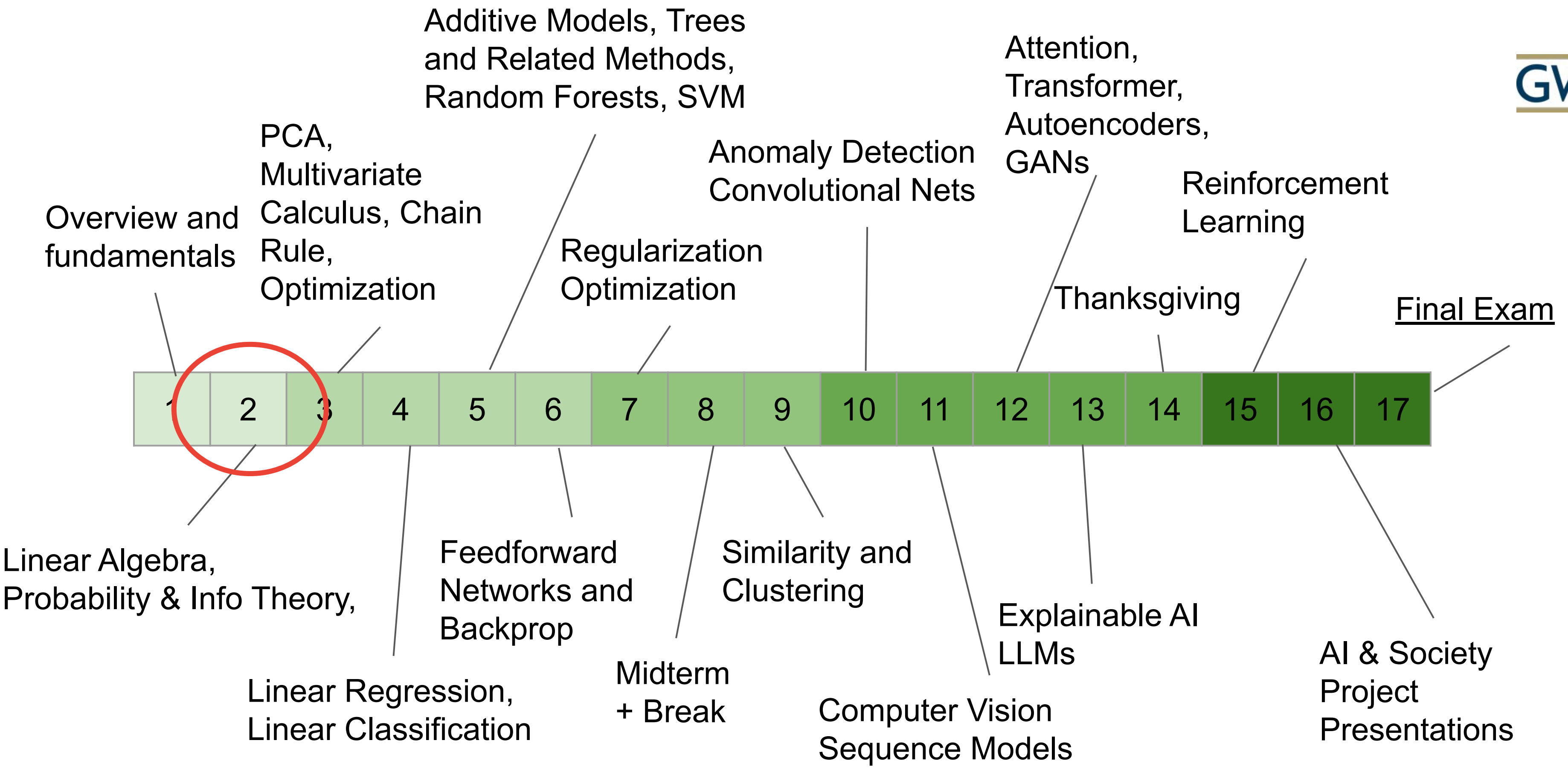
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

CS 4364/6364

Machine Learning

Fall Semester 8/29/2023
Lecture 2.
Linear Algebra Review + Principal Components Analysis

John Sipple
jsipple@gwu.edu



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Overview and fundamentals
Linear Algebra, Probability & Info Theory,
PCA, Multivariate Calculus, Chain Rule, Optimization
Linear Regression, Linear Classification
Additive Models, Trees and Related Methods, Random Forests, SVM
Feedforward Networks and Backprop
Regularization Optimization
Midterm + Break
Similarity and Clustering
Anomaly Detection Convolutional Nets
Computer Vision Sequence Models
Attention, Transformer, Autoencoders, GANs
Explainable AI LLMs
Thanksgiving
Reinforcement Learning
AI & Society Project Presentations
Final Exam

Homework 1

Homework 1

Due date: 9/12/2023

Familiarization with the environment:

- Python Language and Programming Style Guide
- ML Libraries: Tensorflow, Keras, Scikit-Learn
- Google Colaboratory Notebook
- Tensorboard

Training and Evaluating Binary Classifiers

- Cross-fold validation

Hyperparameter Tuning

Comparing Linear Regression against Neural Network

Review of Linear Algebra

- High-level refresher
- Focused on the most important parts for machine learning
- Recommend dusting off your books on Linear Algebra, Calculus, and Probabilities

Scalars

- A single number
- Integers, real numbers, rational numbers
- We'll denote them with an italic:

$$a, u, d$$

Vectors

- A vector is a 1-D array of numbers:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Real-valued in dimension n: $\mathbb{R}^n$

- Integer/binary in dimension n: $\mathbb{Z}^n$

Matrices

- A 2-D array of numbers:

$$\mathbf{A} = \begin{array}{c} \begin{array}{cc} \text{column} & \phantom{A_{1,2}} \\ \downarrow & \end{array} \\ \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \leftarrow \text{row} \\ \\ \end{array}$$

- Example notation for type and shape:

$$\mathbf{A} \in \mathbb{R}^{m \times n}$$

Tensors

A tensor is an array of numbers that may have

- Zero dimensions → scalar
- One dimensional → vector
- Two dimensions → matrix
- And any number of dimensions…

Matrix Transpose

$$(\mathbf{A}^\top)_{i.j} = A_{j,i}$$

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \Rightarrow \boldsymbol{A}^\top = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$$

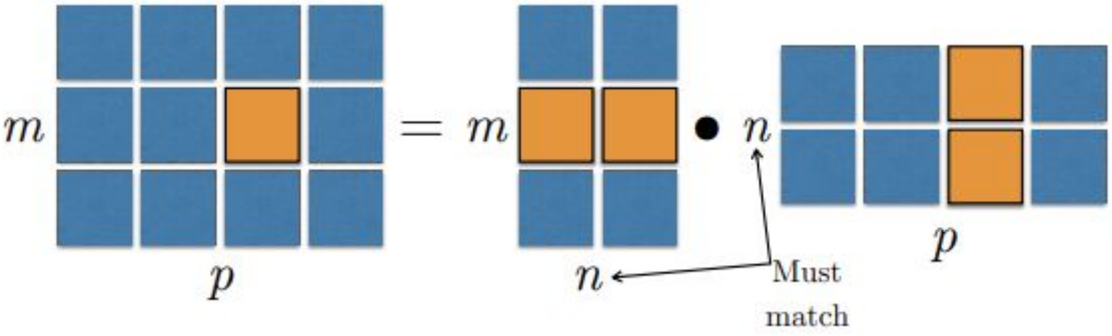
Figure 2.1: The transpose of the matrix can be thought of as a mirror image across the main diagonal.

$$(\mathbf{AB})^\top = \mathbf{A}^\top \mathbf{B}^\top$$

Matrix (Dot) Product

$$\mathbf{C} = \mathbf{A}\mathbf{B}$$

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$



Identity Matrix

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & & & & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$\mathbf{I}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\forall x \in \mathbb{R}^n,$$
$$I_n\mathbf{x} = x$$

Systems of Equations

$$\mathbf{A}\boldsymbol{x} = \mathbf{b}$$

Expands to:

$$\mathbf{A}_{1,:}\boldsymbol{x} = b_1$$
$$\mathbf{A}_{2,:}\boldsymbol{x} = b_2$$
$$\cdots$$
$$\mathbf{A}_{m,:}\boldsymbol{x} = b_m$$

Solving systems of equations

A linear system of equations can have:

- No solution (Underdetermined)
- Many solutions (Overdetermined)
- Exactly one solution → multiplication by the matrix is an invertible function

Matrix Inversion

Matrix inverse:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$$

Solving a system using an inverse:

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{I}_n\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

Numerically unstable, but useful for abstract analysis

Matrix Invertibility

A Matrix cannot be inverted if

- More rows than columns
- More columns than rows
- Redundant rows/columns (linearly dependent or low rank)

Norms

Functions measure how large a vector wrt the origin

Similar to a distance between zero and the point represented by a vector (i.e., distance from zero)

$$f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = \mathbf{0}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$$

Triangle inequality

$$\forall \alpha \in \mathbb{R}, f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$$

Norms

L^p norm (*Minkowski* norm):

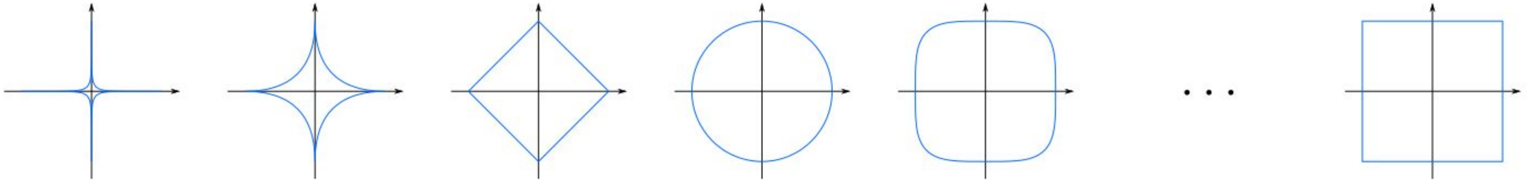
$$||\mathbf{x}||_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

Most popular norm: $L2$ *Euclidean*, $p = 2$

$L1$ City Block norm

$$p = 1 : ||\mathbf{x}||_i = \sum_i |x_i|$$

Max norm $L_\infty : ||\mathbf{x}|| = \max_i |x_i|$



$p = 2^{-2} = 0.25$ | $p = 2^{-1} = 0.5$ | $p = 2^0 = 1$ | $p = 2^1 = 2$ | $p = 2^2 = 4$ | $\cdots$ | $p = 2^\infty = \infty$

https://en.wikipedia.org/wiki/Minkowski_distance

Frobenius Norm

How large the values of a matrix are:

$$||A||_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

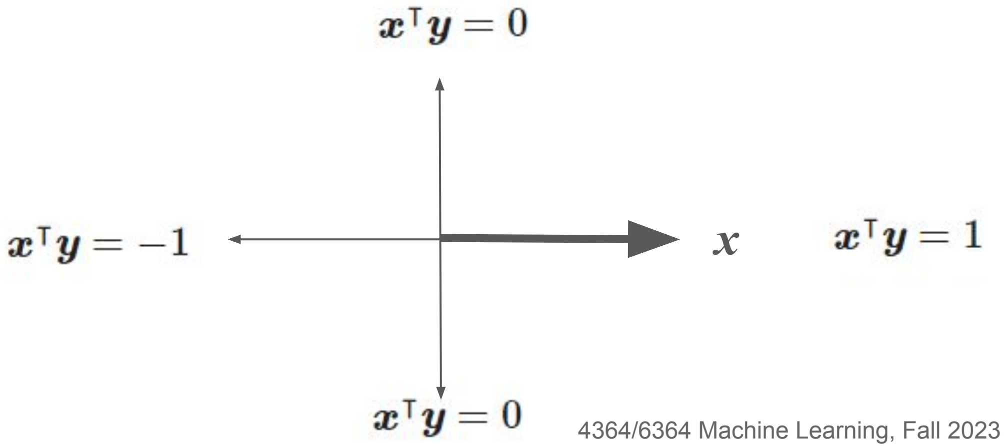
⇒ If A is an error matrix, Frobenius norm is the overall error value which we want to minimize

Dot product

The dot product of two vectors ***x, y*** can be written in terms of norms:

$$\boldsymbol{x}^\top \boldsymbol{y} = ||\boldsymbol{x}||_2 \, ||\boldsymbol{y}||_2 \cos\theta$$

Where θ is the angle between ***x, y***:



Special Matrices and Vectors

Unit vector:

$$||x||_2 = 1$$

Symmetric Matrix:

$$\mathbf{A} = \mathbf{A}^\intercal$$

Orthogonal Matrix:

$$\mathbf{A}^\intercal \mathbf{A} = \mathbf{A} \mathbf{A}^\intercal = \mathbf{I}$$

$$\mathbf{A}^{-1} = \mathbf{A}^\intercal$$

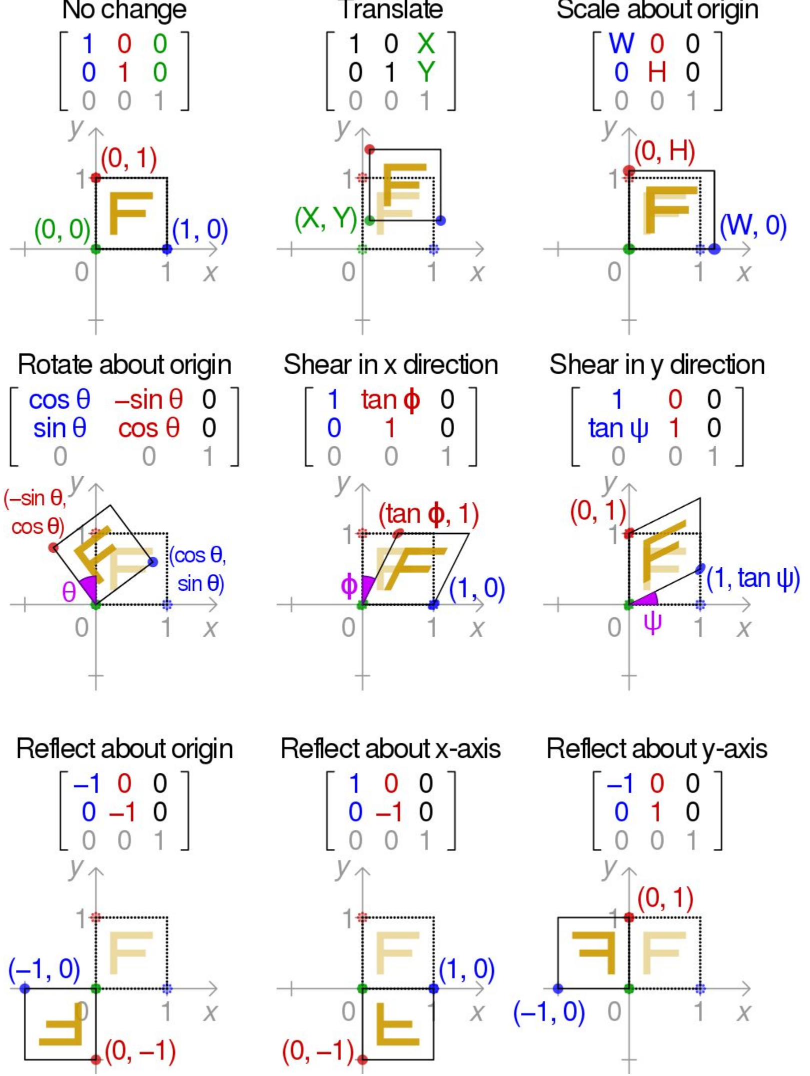
Affine Transformations

Linear matrix transformations that projects

- Points to points
- Lines to lines
- Hyperplanes to hyperplanes

Identity, Translation, Scale, Rotate, Shear and Reflection

A product of one or more affine transformations is itself an affine transformation



Eigendecomposition

Eigenvector $\boldsymbol{v}$ and eigenvalue λ

$$\mathbf{A}\boldsymbol{v} = \lambda\boldsymbol{v}$$

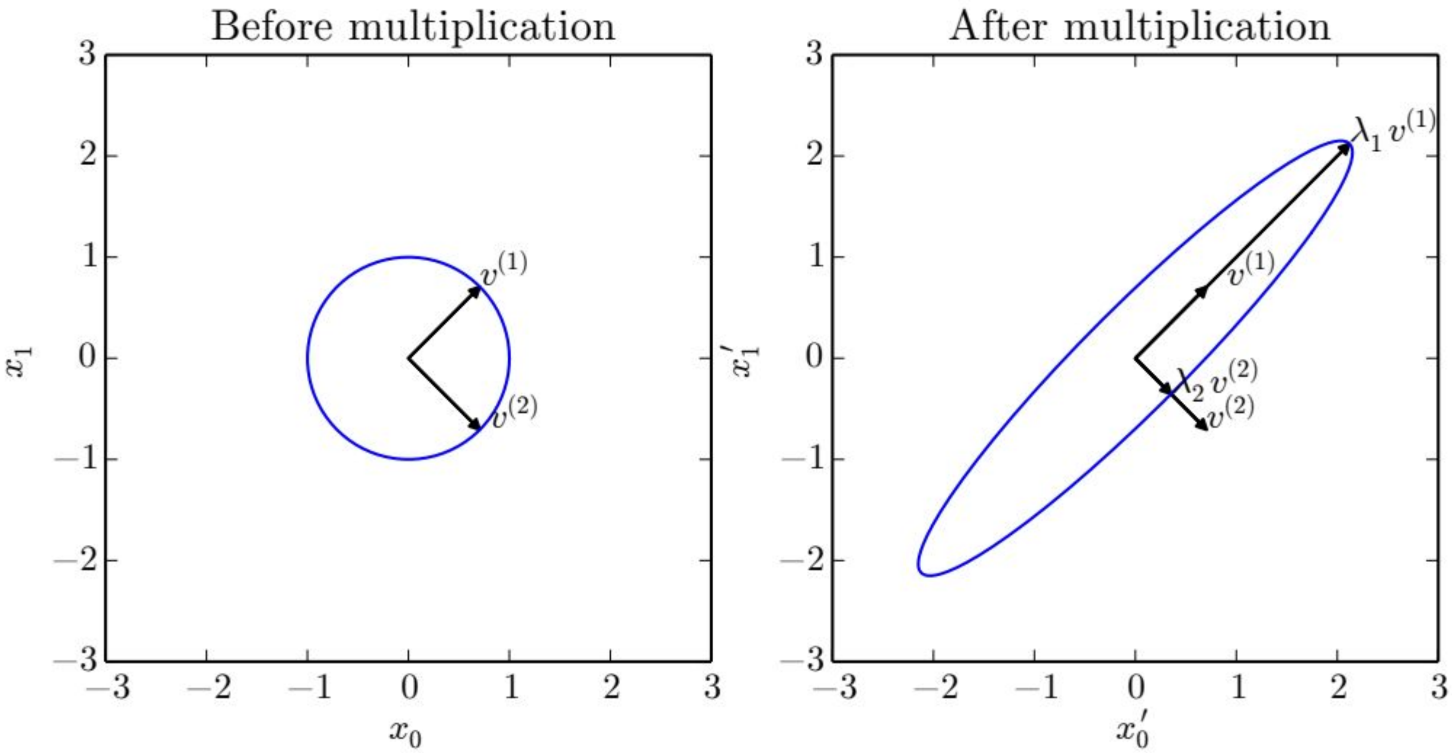
Eigendecomposition of a diagonalizable matrix:

$$\mathbf{A} = \mathbf{V}\mathrm{diag}(\lambda)\mathbf{V}^{-1}$$

Every real symmetric matrix has a real, orthogonal eigendecomposition:

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^{\top}$$

Scaling effect of Eigenvalues



Matrix Terminology

Singular Matrix: any eigenvalue is zero (i.e., pancake)

Positive Definite Matrix: All eigenvalues are positive

Positive Semidefinite Matrix: All eigenvalues are 0 or positive

N**egative Definite Matrix**: All eigenvalues are negative

Negative Semidefinite Matrix: All eigenvalues are 0 or negative

Singular Value Decomposition

Similar to eigendecomposition

More general – matrix need not be square

$\mathbf{D}$: Singular Values (diag)
$\mathbf{U}$: Left-Singular Vectors (orthogonal)
$\mathbf{V}$: Right-Singular Vectors (orthogonal)

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m}\,\underbrace{\mathbf{D}}_{m \times n}\,\underbrace{\mathbf{V}^\top}_{n \times n}$$

Moore-Penrose Pseudoinverse

$$\boldsymbol{x} = \mathbf{A}^{+}\boldsymbol{y}$$

If the equation has:

- Exactly one solution: same as the inverse
- No solution: this gives us the solution with the smallest error $||\mathbf{A}\boldsymbol{x} - \boldsymbol{y}||_2$
- Many solutions: this gives us the solution with the samples norm of $\boldsymbol{x}$

Computing the pseudoinverse

The SVD allows for computing the pseudoinverse

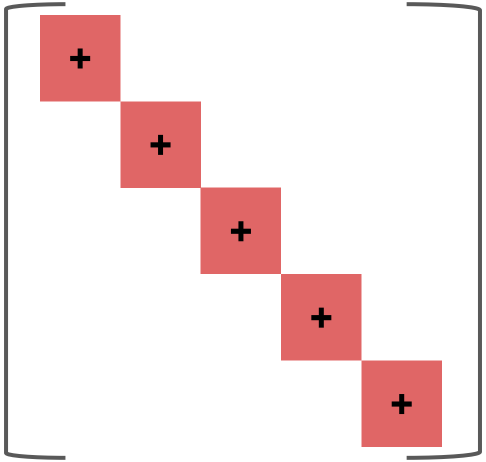
$$\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^\top$$

Take the reciprocal of the nonzero entries and the transpose from $\mathbf{D}$ in:

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

Matrix Trace

Sum of the diagonal elements of matrix $\mathbf{A}$



$$\mathrm{Tr}(\mathbf{A}) = \sum_i \mathbf{A}_{i,i}$$

$$\mathrm{Tr}(\mathbf{ABC}) = \mathrm{Tr}(\mathbf{CAB}) = \mathrm{Tr}(\mathbf{BCA})$$

$$\mathrm{Tr}(\mathbf{A} + \mathbf{B} + \mathbf{C}) = \mathrm{Tr}(\mathbf{A}) + \mathrm{Tr}(\mathbf{B}) + \mathrm{Tr}(\mathbf{C})$$

$$||\mathbf{A}||_F = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^\intercal)}$$

Matrix Determinant

Product of all the eigenvalues $\det(\boldsymbol{A})$ for square matrix $\boldsymbol{A}$

Scalar measure of expansion/contraction

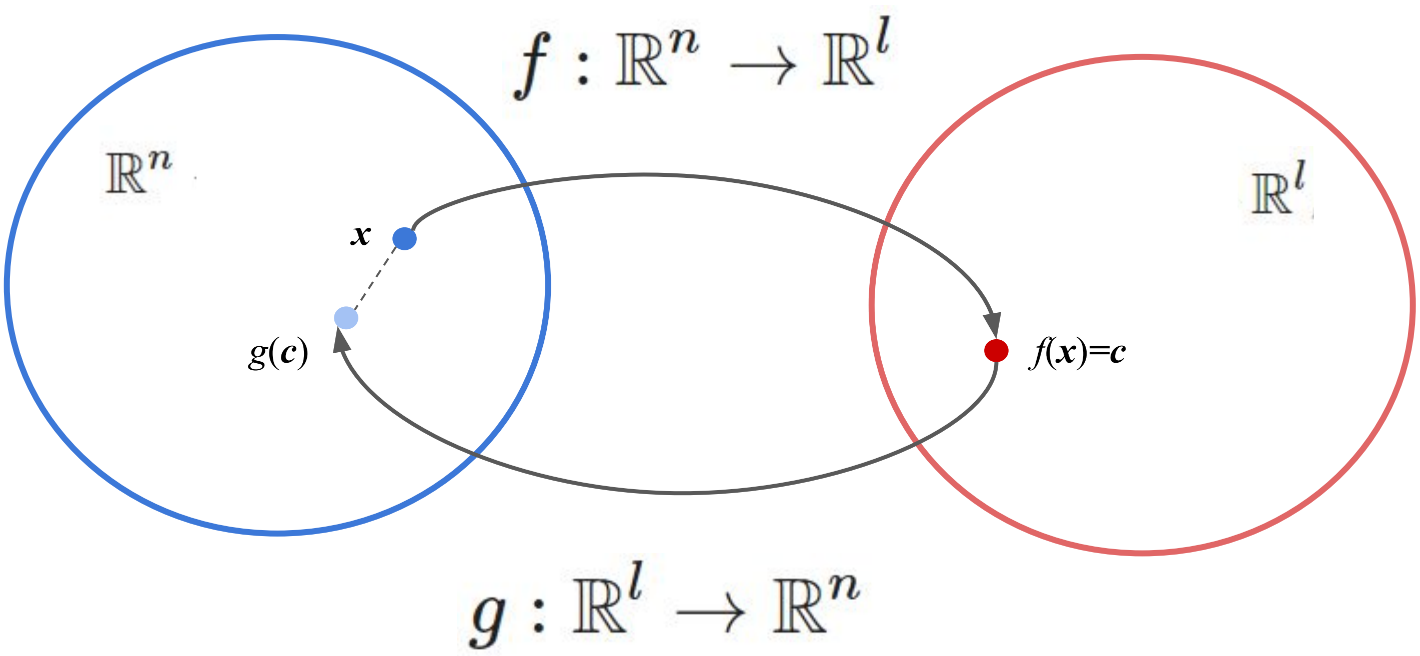
If $\det(\boldsymbol{A}) = 0$ - singular matrix, where at least one dim is 0

If $\det(\boldsymbol{A}) = 1$ - preserves volume

Principal Components Analysis

- Suppose we have m points $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ in $\mathbb{R}^n$
- We'd like to find a lower dimensional mapping: $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^l$, where $\boldsymbol{l} < \boldsymbol{n}$
- For every instance $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$, there is a corresponding code $\boldsymbol{c}^{(i)}$
- **Encoding function** $f(\boldsymbol{x}) = \boldsymbol{c}$
- **Decoding function** $g(\boldsymbol{c}) = r(\boldsymbol{x})$, and $r(\boldsymbol{x}) = g(f(\boldsymbol{x})) \approx \boldsymbol{x}$
- Any useful application come to mind?

Principal Components Analysis



Principal Components Analysis

- Let's choose a very simple decoder based on matrix multiplication

$$g(\boldsymbol{c}) \equiv \mathbf{D}\boldsymbol{c}$$

  where $\mathbf{D} \in \mathbb{R}^{n \times l}$
- Let's add the following constraints (*orthonormal basis*):

1. Columns of $\mathbf{D}$ are orthogonal to each other.
2. Columns of $\mathbf{D}$ have unit norm.

Principal Components Analysis

Need to choose the optimal code point $\boldsymbol{c}^*$ for any input $\boldsymbol{x}$

- Minimize the distance *reconstructon loss* between $\boldsymbol{x}$ and $g(\boldsymbol{c}^*)$ using the L^2 norm:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} ||\boldsymbol{x} - g(\boldsymbol{c})||_2$$

- Is equivalent to minimizing the squared L^2 norm:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} ||\boldsymbol{x} - g(\boldsymbol{c})||_2^2$$

- By defintion of the L^2 norm, function simplifies to:

$$\begin{aligned}
&(\boldsymbol{x} - g(\boldsymbol{c}))^\top (\boldsymbol{x} - g(\boldsymbol{c})) \\
&= \boldsymbol{x}^\top \boldsymbol{x} - \boldsymbol{x}^\top g(\boldsymbol{c}) - g(\boldsymbol{c})^\top \boldsymbol{x} + g(\boldsymbol{c})^\top g(\boldsymbol{c}) \\
&= \boldsymbol{x}^\top \boldsymbol{x} - 2\boldsymbol{x}^\top g(\boldsymbol{c}) + g(\boldsymbol{c})^\top g(\boldsymbol{c}) \qquad (2.57)
\end{aligned}$$

Principal Components Analysis

- We can drop the first term $\boldsymbol{x}^\top \boldsymbol{x}$ in (2.57) since it doesn't depends on $\boldsymbol{c}$

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top g(\boldsymbol{c}) + g(\boldsymbol{c})^\top g(\boldsymbol{c})$$

- Substitute in the defintion of $g(\boldsymbol{c}) = \mathbf{D}\boldsymbol{c}$:

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \mathbf{D}\boldsymbol{c} + \boldsymbol{c}^\top \mathbf{D}^\top \mathbf{D}\boldsymbol{c}$$

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \mathbf{D}\boldsymbol{c} + \boldsymbol{c}^\top \mathbf{I}_l \boldsymbol{c}$$

(since D is orthogonal and unit norm)

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} -2\boldsymbol{x}^\top \mathbf{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{c}$$

Principal Components Analysis

- Using **vector calculus** we can replace $\arg\min_{\boldsymbol{c}}$ with gradient $\nabla(\cdot) = \mathbf{0}$:

$$\nabla_{\boldsymbol{c}}\left(-2\boldsymbol{x}^\top \mathbf{D}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{c}\right) = \mathbf{0}$$

$$-2\mathbf{D}^\top \boldsymbol{x} + 2\boldsymbol{c} = \mathbf{0}$$

$$\boldsymbol{c} = \mathbf{D}^\top \boldsymbol{x}$$

- We can optimally encode $\boldsymbol{x}$ with just matrix-vector operation!

$$f(\boldsymbol{x}) = \mathbf{D}^\top \boldsymbol{x}$$

- PCA reconstruction operation:

$$r(\boldsymbol{x}) = g(f(\boldsymbol{x})) = \mathbf{D}\mathbf{D}^\top \boldsymbol{x}$$

Principal Components Analysis

- We want to choose $\mathbf{D}$ that minimizes the reconstruction error $\boldsymbol{x} - r(\boldsymbol{x})$ for all m points
- Apply the **Frobenius norm** of the error matrix $\mathbf{X} - r(\mathbf{X})$ for all n dimensions and m points:

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sqrt{\sum_{i,j} \left(x_j^{(i)} - r(\boldsymbol{x}^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_l \tag{2.68}$$

Principal Components Analysis

- To get to $\mathbf{D}^*$, let's start by considering one-dimensional projection, $l = 1$, and later expand to $l > 1$
- This makes $\mathbf{D}$ just a 1-dimensional matrix (i.e., vector) $\boldsymbol{d}$ simplifying (2.68):

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i ||x^{(i)} - \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{x}^{(i)}||_2^2 \text{ subject to } ||\boldsymbol{d}||_2 = 1$$

- Rearrange the terms into standard formatting, noting that a scalar and its transpose are equal:

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i ||x^{(i)} - \boldsymbol{d}^\top \boldsymbol{x}^{(i)} \boldsymbol{d}||_2^2 \text{ subject to } ||\boldsymbol{d}||_2 = 1$$

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \sum_i ||x^{(i)} - \boldsymbol{x}^{(i)\top} \boldsymbol{d}\boldsymbol{d}||_2^2 \text{ subject to } ||\boldsymbol{d}||_2 = 1$$

Principal Components Analysis

- Now, we'll rewrite this in terms of the design matrix: $\mathbf{X} = \left[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\right]^\top \in \mathbb{R}^{m \times n}$

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} ||\mathbf{X} - \mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top||_F^2 \text{ subject to } \boldsymbol{d}^\top\boldsymbol{d} = 1$$

- Ignoring the constraint for a moment, and focusing on the Frobenius norm:

$$\arg\min_{\boldsymbol{d}} ||\mathbf{X} - \mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top||_F^2$$

$$= \arg\min_{\boldsymbol{d}} \operatorname{Tr}\left(\left(\mathbf{X} - \mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top\right)^\top\left(\mathbf{X} - \mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top\right)\right) \quad (2.74)$$

Principal Components Analysis

- Rewriting (2.74):

$$= \arg\min_{\boldsymbol{d}} \operatorname{Tr}\left(\mathbf{X}^\top\mathbf{X} - \mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top - \boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X} + \boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top\right)$$

- Trace of a sum is the sum of the traces:

$$= \arg\min_{\boldsymbol{d}} \operatorname{Tr}(\mathbf{X}^\top\mathbf{X}) - \operatorname{Tr}(\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top) - \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}) + \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top)$$

- Drop $\operatorname{Tr}(\mathbf{X}^\top\mathbf{X})$ because it doesn't affect $\boldsymbol{d}$:

$$= \arg\min_{\boldsymbol{d}} -\operatorname{Tr}(\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top) - \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}) + \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top)$$

- We can rearrange order of a matrix product inside trace:

$$= \arg\min_{\boldsymbol{d}} -2\operatorname{Tr}(\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top) + \operatorname{Tr}(\boldsymbol{d}\boldsymbol{d}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top)$$

$$= \arg\min_{\boldsymbol{d}} -2\operatorname{Tr}(\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top) + \operatorname{Tr}(\mathbf{X}^\top\mathbf{X}\boldsymbol{d}\boldsymbol{d}^\top\boldsymbol{d}\boldsymbol{d}^\top)$$

Principal Components Analysis

- Now, let's bring the constraint back and apply it to simplify further:

$$= \arg\min_{\boldsymbol{d}} -2\mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top) + \mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top \boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

$$= \arg\min_{\boldsymbol{d}} -2\mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top) + \mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

- Adding two identical terms:

$$= \arg\min_{\boldsymbol{d}} -\mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

- Drop the minus and make a maximization problem:

$$= \arg\max_{\boldsymbol{d}} \mathrm{Tr}(\mathbf{X}^\top \mathbf{X} \boldsymbol{d}\boldsymbol{d}^\top) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

- Rearrange terms inside the Trace:

$$= \arg\max_{\boldsymbol{d}} \mathrm{Tr}(\boldsymbol{d}^\top \mathbf{X}^\top \mathbf{X} \boldsymbol{d}) \text{ subject to } \boldsymbol{d}^\top \boldsymbol{d} = 1$$

Principal Components Analysis

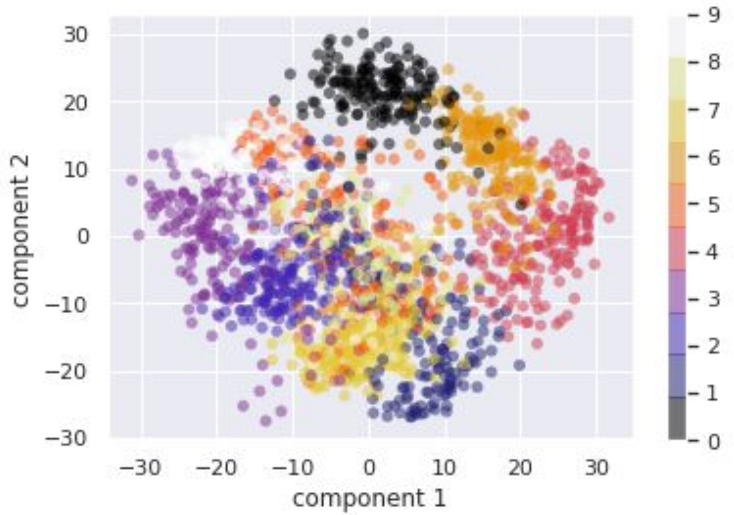
- The optimization problem is solved via eigendecomposition
- The optimal $\boldsymbol{d}$ is given by the eigenvector of $\mathbf{X}^\top\mathbf{X}$ corresponding to the largest eigenvalue: $\max(\Lambda)$:

$$\mathbf{X}^\top\mathbf{X} = \mathbf{Q}\Lambda\mathbf{Q}^\top$$

- For $l > 1$, choose $\mathbf{D}$ using the eigenvectors in $\mathbf{Q}$ corresponding to the l largest eigenvalues
- Can be proven via induction.

PCA Demo

Source:
https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.09-Principal-Component-Analysis.ipynb#scrollTo=NKCCssS-tNR-



Great colab tutorial on linear algebra

https://github.com/jonkrohn/ML-foundations/blob/master/notebooks/2-linear-algebra-ii.ipynb

Discussion about your final project