Tsinghua-Berkeley Shenzhen Institute

Learning From Data

Fall 2020

Review Session 1

Weida Wang, Feng Zhao — September 20, 2020

Aim: This note is to review some basic mathematical knowledge on linear algebra, calculus and probability, and to introduce some scientific programming background in Python. We hope it can assist you in your future coursework.

1 Linear Algebra

1.1 Inner Product and trace

Definition 1. (Inner product). A function $\langle \cdot, \cdot \rangle$: $\mathbb{V} \times \mathbb{V} \to \mathbb{F}$ is an inner product if it satisfies [1]:

- **Linearity**: $\langle \alpha \boldsymbol{v} + \beta \boldsymbol{w}, \boldsymbol{x} \rangle = \alpha \langle \boldsymbol{v}, \boldsymbol{x} \rangle + \beta \langle \boldsymbol{w}, \boldsymbol{x} \rangle$;
- **Conjugate symmetry**: $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \overline{\langle \boldsymbol{w}, \boldsymbol{v} \rangle}$;
- **Positive definiteness**: $\langle \boldsymbol{v}, \boldsymbol{v} \rangle \geq 0$, with the equality iff $\boldsymbol{v} = 0$;

The main example is the canonical inner product on $\mathbb{R}^n$, which simply sets

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = x_1 y_1 + x_2 y_2 + \cdots x_n y_n = \sum_{i=1}^{n} x_i y_i.$$

And we say vector $\boldsymbol{x} \in \mathbb{R}^n$ is orthogonal to $\boldsymbol{y} \in \mathbb{R}^n$ when $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$.

The $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is said to be **orthogonal** if its columns are pairwise orthogonal, which implies that

$$\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = I$$

Definition 2. (Trace). For $\mathbf{M} \in \mathbb{R}^{n \times n}, \operatorname{trace}(\mathbf{M}) = \sum\limits_{i=1}^{n} \mathbf{M}_{ii}$, where $\mathbf{M}_{ii}$ is the diagonal terms of matrix $\mathbf{M}$.

Theorem 1. For any matrices $\mathbf{A}, \mathbf{B}$ of compatible size,

$$\operatorname{trace}(\mathbf{AB}) = \operatorname{trace}(\mathbf{BA})$$

1.2 Eigenvalue Decomposition

Definition 3. (Eigenvalue, eigenvector). Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$. We sat that $\lambda \in \mathbb{C}$ is an eigenvalue of $\boldsymbol{A}$ if there exists some nonzero vector $\boldsymbol{v} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$ such that

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

Theorem 2. (Eigenvector decomposition). Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be symmetric. Then there exist orthonormal vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$ and real scalars $\lambda_1 \geq \cdots \geq \lambda_n$, such that if we write

$$\boldsymbol{V} = [\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_n] \in O(n), \quad \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \in \mathbb{R}^{n\times n},$$

we have

$$\boldsymbol{A} = \boldsymbol{V}\Lambda\boldsymbol{V}^*.$$

The expression $\boldsymbol{A} = \boldsymbol{V}\Lambda\boldsymbol{V}^*$ can also be written as $\boldsymbol{A} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^*$.

1.3 Vector and Matrix Norms

A norm on a vector space $\mathbb{V}$ gives a way of measuring lengths of vectors. Formally:

Definition 4. (Vector norm). A norm on a real vector space $\mathbb{V}$ is a function $||\cdot|| : \mathbb{V} \to \mathbb{R}$ that is:

- **Nonnegatively homogeneous**: $||\alpha \boldsymbol{x}|| = |\alpha| ||\boldsymbol{x}||$ for all vectors $\boldsymbol{x} \in \mathbb{V}$, scalars $\alpha \in \mathbb{R}$;
- **Positive definite**: $||\boldsymbol{x}|| \geq 0$, and $||\boldsymbol{x}|| = 0$ iff $\boldsymbol{x} = 0$;
- **Subadditive**: $||\cdot||$ satisfies the triangle inequality $||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.

One very important family of norms are the ℓ^p norms. If we take $\mathbb{V} = \mathbb{R}^n$, and $p \in [1, \infty)$, we can write

$$||\boldsymbol{x}||_p = \left(\sum_i |\boldsymbol{x}_i|^p \right)^{\frac{1}{p}}. \tag{1}$$

The most familiar example is the ℓ^2 norm or the "Euclidean norm"

$$||\boldsymbol{x}||_2 = \sqrt{\sum_i \boldsymbol{x}_i^2} = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$$

which coincides with our usually way of measuring lengths. Two other cases are of almost equal importance: $p = 1$, and $p \to \infty$. Setting $p = 1$ in (1), we obtain

$$||\boldsymbol{x}||_1 = \sum_i |\boldsymbol{x}_i|.$$

Finally, as p becomes larger, the expression in (1) accentuates large $|\boldsymbol{x}_i|$. As $p \to \infty$, $||\boldsymbol{x}||_p \to \max_i |\boldsymbol{x}_i|$. Thus, we can extend the definition of the ℓ^p norm to $p = \infty$ by defining

$$||\boldsymbol{x}||_\infty = \max_i |\boldsymbol{x}_i|.$$

Likewise, we can give the norm of matrices which characterizes the feature of matrices.

Definition 5. (Matrix norm). A norm on a real matrix space $\mathbb{R}^{m\times n}$ is a function $||\cdot|| : \mathbb{R}^{m\times n} \to \mathbb{R}$ that is:
- **Nonnegatively homogeneous**: $||\alpha \boldsymbol{A}|| = |\alpha|||\boldsymbol{A}||$ for all matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, scalars $\alpha \in \mathbb{R}$;
- **Positive definite**: $||\boldsymbol{A}|| \geq 0$, and $||\boldsymbol{A}|| = 0$ iff $\boldsymbol{A} = \boldsymbol{0}$;
- **Subadditive**: $||\cdot||$ satisfies the triangle inequality $||\boldsymbol{A} + \boldsymbol{B}|| \leq ||\boldsymbol{A}|| + ||\boldsymbol{B}||$, for all $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$;
- **Submultiplicativity**: $||\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B}|| \leq ||\boldsymbol{A}||||\boldsymbol{B}||$.

The most famous matrix norms are Frobenius norm and spectral norm which are given by

$$\begin{aligned} ||\boldsymbol{A}||_F &= \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{\frac{1}{2}} = \left(\mathrm{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{m\bigwedge n} \lambda_i\right)^{\frac{1}{2}} \\ ||\boldsymbol{A}||_s &= \sqrt{\lambda_{max}}, \end{aligned}$$

in which λ_{max} is the largest eigenvalue of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.

2 Calculus

2.1 Derivatives

For scalar b, vectors $\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{y}$ and matrix $\boldsymbol{A}$, we have [2]:
- $\dfrac{\partial(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)}{\partial \boldsymbol{x}} = \boldsymbol{w}$
- $\dfrac{\partial(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + b)}{\partial \boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}$
- $\dfrac{\partial(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{y})}{\partial \boldsymbol{A}} = -\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{-\mathrm{T}}$

3 Probability

3.1 Basic Properties

For events E_1 and E_2, if they are disjoint, i.e. $E_1 \cap E_2 = \emptyset$, then $\mathbb{P}(E_1 \cup E_2) = \mathbb{P}(E_1) + \mathbb{P}(E_2)$

Definition 6. (Conditional probability) For events A and B, and $\mathbb{P}(A) > 0$,

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}$$

We can define the conditional expectation as

$$\mathbb{E}\left[Y|X=x\right] \triangleq \sum_{y \in \mathcal{Y}} y \cdot p\left(Y=y|X=x\right)$$

For two random variables X and Y, the covariance is defined by

$$\mathrm{Cov}\left[X, Y\right] = \mathbb{E}\left[XY\right] - \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right]$$

When the covariance of X and Y is 0, we call them uncorrelated variables.

For two random variables, when the joint pdf can be written as the product of two RVs' pdf

$$f\left(x, y\right) = f_X\left(x\right) f_Y\left(y\right),$$

we call them independent.

Theorem 3. We have:

∘ (Multiplication Rule) For events A and B,

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B|A) = \mathbb{P}(B)\mathbb{P}(A|B);$$

∘ (Total probability rule) $B_1, B_2, \ldots, B_k$ form a partition of Ω, $\forall i \neq j, B_i \cap B_j = \emptyset, \cup_{i=1}^{k} B_i = \Omega$, we have:

$$\mathbb{P}(A) = \sum_{i=1}^{k} \mathbb{P}(B_i)\mathbb{P}(A|B_i);$$

∘ (Bayes Rule)

$$\mathbb{P}(B_1|A) = \frac{\mathbb{P}(A \cap B_1)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A|B_1)\mathbb{P}(B_1)}{\mathbb{P}(A)} = \frac{\mathbb{P}(A|B_1)\mathbb{P}(B_1)}{\sum_{i=1}^{k} \mathbb{P}(A|B_i)\mathbb{P}(B_i)}.$$

3.2 Gaussian Distribution

3.2.1 Normal Distribution

- If random variable $X \in \mathbb{R}$, $X \sim \mathcal{N}(\mu, \sigma^2)$, then the density function of it is:

$$p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- $\mathbb{E}[X] = \mu$; $\mathrm{var}(X) = \sigma^2$.

3.2.2 Multivariate Gaussian Distribution

- If random variable $\boldsymbol{X} \in \mathbb{R}^n$, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the density function of it is:

$$p(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$; $\mathrm{cov}(\boldsymbol{X}) = \boldsymbol{\Sigma}$.

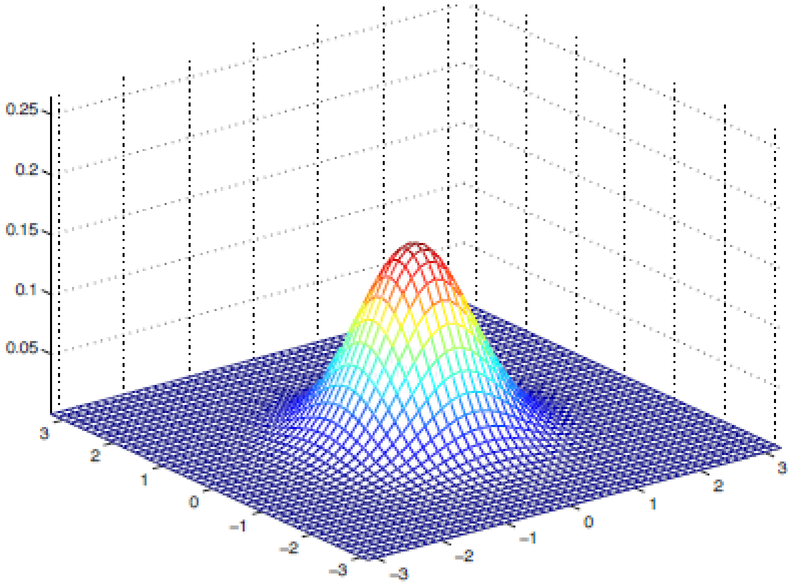

Figure 1: Multivariate Gaussian's p.d.f

4 Scientific Programming in Python

Python is a general purpose programming language. With the help of some powerful packages, it is possible to do scientific computing with Python. In this section, a short introduction on how to do scientific programming is given. To be more specific, we will introduce how to use `numpy`, `scipy` and `matplotlib` to do linear algebra, statistics and basic 2D plotting.

4.1 Numpy – n-dimensional array manipulation

The package `numpy` provides a convenient way for operations on n-dimensional array. As a special case, the vector is 1-dimensional array while the matrix is called 2-dimensional array. These two structures are the most commonly used in practice. The following code shows how to create a vector of length 3 and compute its ℓ_2 norm.

```python
import numpy as np
a = np.array([1, 2, 3])
print(np.linalg.norm(a))
```

As is noticed, most methods related with linear algebra are under the module prefix `np.linalg`. We give another example to compute the eigenvalues of a square matrix:

```python
A = np.array([[1, 2], [3, 4]])
print(np.linalg.eig(A)[0])
```

The function `np.linalg.eig` returns two values, the first is the eigenvalues and the second is the eigenvectors. Therefore, we use the index `[0]` to retrieve only the first return value in the above code. For detailed documentation, you can use `help(np.linalg.eig)`. There are some advanced techniques to manipulate two-dimensional array. For example, if we want to compute the summation of each row for a matrix. we can provide an optional parameter `axis=1` as follows:

```python
A = np.array([[1, 2], [3, 4], [5, 6]])
print(np.sum(A, axis=1))
```

Notice that `A` is a 3×2 matrix. After the operation of the **reduced sum** on `A`, we get a vector. Another way to add together each row of a matrix is to use matrix product. Contrary to the mathematical representation, in `numpy`, `A * B` is the element-wise multiplication while `A @ B` is the matrix product. Therefore, we need the following code to achieve our goal:

```python
print(A @ np.array([1, 1]))
```

4.2 Scipy – algorithms of applied mathematics

The package `scipy` provides many useful algorithms on various domains. In this subsection, we focus on the subpackage `scipy.stats`.

You have already known that a continuous random variable has its probability density function (pdf). Theses functions are available in `scipy.stats`. For example, the pdf of normal distribution can be queried by:

```
import scipy.stats
x = np.linspace(-3, 3)
y = scipy.stats.norm.pdf(x)
print(x, y)
```

In this code snippet, we have used the function `np.linspace` to generate a vector of length 50. As the function name suggests, the distances between adjacent numbers are equal. Then we apply the function `scipy.stats.norm.pdf` to this vector. It should be noticed that `numpy` and `scipy` functions can be applied to array directly. In logic, it is equivalent to apply each element of the array and combine the result to an array. But it is much faster to use array as the input parameter instead of writing a for-loop by hand.

4.3 Matplotlib – plotting experiment results

The package `matplotlib` provides drawing method to plot figures from data. We illustrate the basic usage of this package by considering a simple experiment. That is, we sample data from a Gaussian distribution, get the empirical distribution by drawing the histogram and compare the histogram with the Gaussian pdf. The code for this procedure is as follows:

```
import matplotlib.pyplot as plt
c = np.random.normal(size=1000)
plt.hist(c, density=True)
plt.plot(x, y)
plt.show()
```

We use the function `np.random.normal` to generate 1000 random samples, which are used to draw the density histogram by `plt.hist`. We also draw the true pdf by `plt.plot`. This plot function can accept two vectors and plot a line on 2D plane. Finally, we use `plt.show` to open the figure window and we can see the plotting results immediately.

4.4 Summary

The above paragraphs only give a basic overview on the scientific packages of Python. They are building-blocks of many machine-learning packages. If you would like to learn more on this topic, I recommend a good web resource written by Justin Johnson [3].

References

[1] Strang, Gilbert, et al. Introduction to linear algebra. Vol. 3. Wellesley, MA: Wellesley-Cambridge Press, 1993.

[2] The Matrix Cookbook http://matrixcookbook.com

[3] https://cs231n.github.io/python-numpy-tutorial/