

# STAT 542 - Statistical Learning

## Homework 1 - Manan Mehta (mananm2)

Due: 08/31/2020

### About Homework 1

The goal of HW1 is to check your prerequisite knowledge. All of this knowledge should be already covered in courses such as Stat 410, Stat 425 and other basic mathematical courses. These concepts and skills will be used extensively in our course.

In addition, this HW also check your basic programming knowledge, such as writing a function, random seed, and also Latex. Please note that you are required to type-in all formulas in Latex form in all future homeworks. Failing to do so will lead to some penalty.

### Basic Calculus

#### 1. Calculate the derivatives:

(a)

$$\frac{d}{dx}e^x = e^x$$

(b)

$$\frac{d}{dx}\log(1+x) = \frac{1}{1+x}$$

(c)

$$\frac{d}{dx}\log(1+e^x) = \frac{e^x}{1+e^x}$$

**2. Taylor Expansion** Let  $f: \mathbb{R} \rightarrow \mathbb{R}$  be a twice differentiable function. Please write down the first three terms of its Taylor expansion at point  $x = 1$ .

The Taylor expansion at  $x = 1$  is:

$$f(1) + \frac{f'(1)}{1!}(x-1) + \frac{f''(1)}{2!}(x-1)^2$$

The general Taylor series at  $x = a$  is:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

**3. Infinite Sum Convergence** For the infinite sum  $\sum_{n=1}^{\infty} \frac{1}{n^\alpha}$ , where  $\alpha$  is a positive real number, give the exact range of  $\alpha$  such that the series converges.

The series converges for  $\alpha > 1$  and diverges for  $0 < \alpha \leq 1$

## Linear Algebra

**1. Eigendecomposition** What is the eigendecomposition of a real symmetric matrix  $A_{n \times n}$ ? Write down one form of that decomposition and explain each term in your formula. Based on these terms, derive  $A^{-1/2}$

The eigendecomposition of a real symmetric matrix is:

$$A = PDP^T$$

where  $P_{n \times n}$  is an orthogonal matrix whose columns are the eigenvectors of  $A$  (we know that: if  $A_{n \times n}$  is a real symmetric matrix,  $A$  has  $n$  distinct eigenvectors that form an orthonormal basis for  $\mathbb{R}^n$ )

and  $D_{n \times n} = \text{diag}(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$

Thus we can find  $A^{-1/2}$  as:

$$A^{-1/2} = PD^{-1/2}P^T$$

where  $D^{-1/2} = \text{diag}(\sqrt{\epsilon_1}, \sqrt{\epsilon_2}, \dots, \sqrt{\epsilon_n})$

**2. SPD Matrix** A matrix  $A_{n \times n}$  is symmetric positive definite if and only if all its eigenvalues are positive. Alternatively,  $A_{n \times n}$  is symmetric positive definite if:

$$x^T Ax > 0 \quad \forall x \in \mathbb{R}^n / 0$$

**3. True or False** For two real matrices  $A_{m \times n}$  and  $B_{n \times m}$

(a)  $\text{Rank}(A) = \max\{m, n\}$

**False** :  $\text{Rank}(A) \leq \min\{m, n\}$  where the equality holds only when  $A$  is full ranked.

(b) If  $m = n$ , then  $\text{trace}(A) = \sum_{i=1}^n A_{ii}$

**True**

(c) If  $A$  is a symmetric matrix, then all eigenvalues of  $A$  are real

**True**

(d) If  $A$  is a symmetric matrix,  $\lambda_1$  and  $\lambda_2$  are two of its eigen-values (not necessarily different) and  $v_1, v_2$  are the corresponding eigen-vectors, then  $v_1^T v_2 = 0$

**True** : Eigenvectors of the eigenvalues of a symmetric matrix are mutually orthogonal

(e)  $\text{trace}(ABAB) = \text{trace}(AABB)$

**False** : If  $A$  is  $m \times n$  and  $B$  is  $n \times m$  then the product  $AABB$  is not valid. Even if  $A$  and  $B$  are  $n \times n$ , trace only remains unchanged under cyclic permutations. Thus  $\text{trace}(ABAB) = \text{trace}(BABA) \neq \text{trace}(AABB)$

## Statistics

**1. Estimators**  $X_1, X_2, \dots, X_n$  are i.i.d.  $\mathcal{N}(\mu, \sigma^2)$  random variables, where  $\mu \in \mathbb{R}$  and  $\sigma > 0$  is finite. Let  $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ .

(a) What is an unbiased estimator? Is  $\bar{X}_n$  an unbiased estimator of  $\mu$ ?

An estimator with zero difference between the estimator's expected value and the parameter true value is *unbiased*.  $\bar{X}_n$  is an unbiased estimator of  $\mu$ .

(b) What is  $E[(\bar{X}_n)^2]$  in terms of  $n, \mu, \sigma$ ?

We know the relationship  $Var(\bar{X}_n) = E[(\bar{X}_n)^2] - [E(\bar{X}_n)]^2$

On rearranging, we get  $E[(\bar{X}_n)^2] = Var(\bar{X}_n) + [E(\bar{X}_n)]^2$

Thus,

$$E[(\bar{X}_n)^2] = \frac{\sigma^2}{n} + \mu^2$$

(c) Give an unbiased estimator of  $\sigma^2$

An unbiased estimator of  $\sigma^2$  is sample variance  $S^2$  given by:

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X}_n)^2}{n - 1}$$

(d) What is a consistent estimator? Is  $\bar{X}_n$  a consistent estimator of  $\mu$ ?

A consistent estimator is an estimator, for which, as the number of data points increases indefinitely, the estimate converges in theory to the parameter. Yes,  $\bar{X}_n$  a consistent estimator of  $\mu$ .

**2. Least Squares Model** Suppose  $X_{p \times 1}$  is a vector of covariates,  $\beta_{p \times 1}$  is a vector of unknown parameters,  $\epsilon$  is the unobserved random noise and we assume the linear model relationship  $y = X^T \beta + \epsilon$ . Suppose we have  $n$  i.i.d. samples from this linear model, and the observed data can be written using the matrix form:  $\mathbf{y}_{n \times 1} = \mathbf{X}_{n \times p} \beta_{p \times 1} + \epsilon_{n \times 1}$

(a) If we want estimate the unknown  $\beta$  using a least square method, what is the objective function  $L(\beta)$  to obtain  $\hat{\beta}$ ?

The objective function can be written as:

$$L(\beta) = (y - X\beta)^T (y - X\beta)$$

(b) What is the solution of  $\hat{\beta}$ ? Represent the solution using the observed data  $\mathbf{y}$  and  $\mathbf{X}_{n \times p}$ . Note that you may assume that  $\mathbf{X}^T \mathbf{X}$  is invertible.

The solution  $[ \min(L(\beta)) ]$  is obtained by taking the partial derivative with each  $\beta_j$ . The final expression is:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Reference: [Linear Least Squares \(Stanford StatWeb\)](#)

## Programming

(Course recommends R, but Python is used here)

**1. Linear Least Squares Function** Use the following code to generate a set of observations  $\mathbf{y}$  and  $\mathbf{X}_{n \times p}$ . Following the previously established formula, Write your own code, instead of using existing functions such as `lm()`, to solve for the least square estimator  $\hat{\beta}$ . If you are asked to add an intercept term  $\beta_0$  into your estimation (even the true  $\beta_0 = 0$  in our data generator), what should you do?

```
import numpy as np
from scipy.stats import norm

def getBeta(y, X):
    return (np.linalg.inv(X.T.dot(X))).dot((X.T).dot(y))

#Generate data
seed = 1
n , p = 100 , 5
true_beta = np.array([1 , 0 , 0 , 1 , -1])
X = norm.rvs(size = (n,p), random_state = seed)
epsilon = norm.rvs(size = n, random_state = seed)

y = X.dot(true_beta) + epsilon

#Get a least squared estimate for beta from the generated data
estimated_beta = getBeta(y , X)

#Compare how the estimate matches the true parameters
print("True Beta:", true_beta)
print("Estimated Beta:", np.round(estimated_beta,2))
```

True Beta: [ 1 0 0 1 -1]

Estimated Beta: [ 0.98 -0.03 -0.08 0.93 -0.96]

If asked to add an intercept, we change  $p$  to 6, and add a **column of ones** to the newly generated  $X$ .

**2. Simulation Study** Perform a simulation study to check the consistency of the sample mean estimator  $\bar{X}_n$ . Please save your random seed so that the results can be replicated by others.

- (a) Generate a set of  $n = 20$  i.i.d. observations from uniform  $(0, 1)$  distribution and calculate the sample mean  $\bar{X}_n$

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import uniform
seed = 1
```

```

n = 20
x = uniform.rvs(size = n, random_state = seed)

sample_mean = np.mean(x)
print("The sample mean is {:.2f}".format(sample_mean))

```

The sample mean is 0.37

(b) Repeat step [a] 1000 times to collect 1000 such sample means and plot them using a histogram.

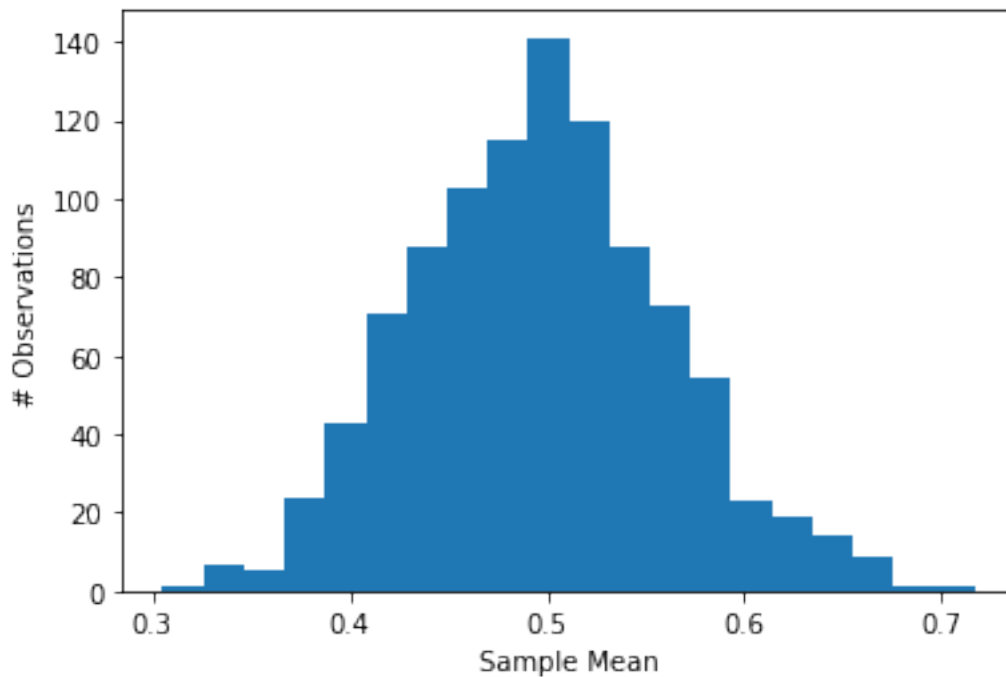
```

#seed in each iteration is the iteration number
n = 20

sample_means = np.zeros(1000)
for i in range(1000):
    sample_means[i] = np.mean(uniform.rvs(size = n, random_state = i))

plt.hist(sample_means, bins = n)
plt.xlabel('Sample Mean')
plt.ylabel('# Observations')
plt.show()

```



(c) How many of such sample means (out of 1000) are at least 0.1 away from true mean parameter, which is 0.5 for uniform (0, 1)?

```
n_outliers = (sample_means < 0.4).sum() + (sample_means > 0.6).sum()
print("Number of sample means at least 0.1 away from 0.5 is {} out of 1000".
      ↪format(n_outliers))
```

Number of sample means at least 0.1 away from 0.5 is 121 out of 1000

(d) Repeat steps [a] to [c] with  $n=100$  and  $n=500$ . What conclusion can you make?

```
#For n = 20
n = 20
sample_means_20 = np.zeros(1000)
for i in range(1000):
    sample_means_20[i] = np.mean(uniform.rvs(size = n, random_state = i))

#For n = 100
n = 100
sample_means_100 = np.zeros(1000)
for i in range(1000):
    sample_means_100[i] = np.mean(uniform.rvs(size = n, random_state = i))

#For n = 500
n = 500
sample_means_500 = np.zeros(1000)
for i in range(1000):
    sample_means_500[i] = np.mean(uniform.rvs(size = n, random_state = i))

#Compare all three side-by-side
fig, axarr = plt.subplots(1,3, figsize = (12,4))
ax1, ax2, ax3 = axarr[0], axarr[1], axarr[2]

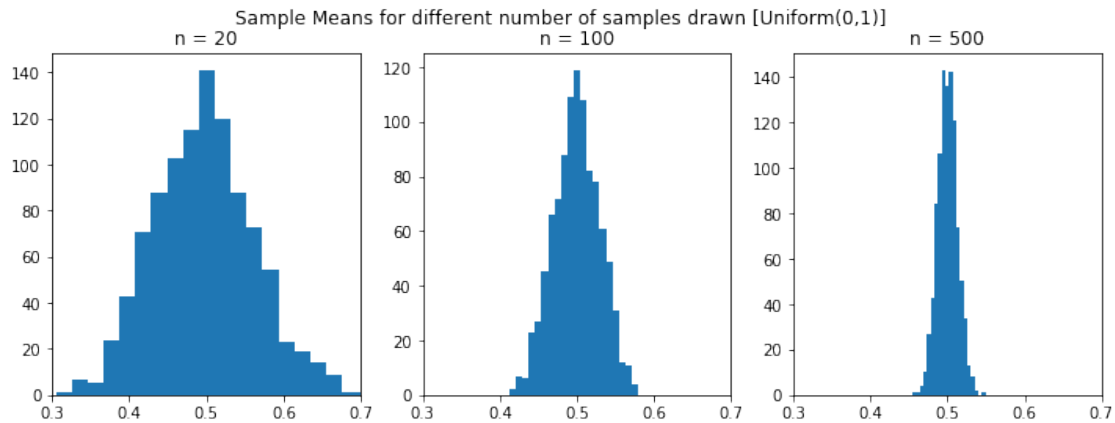
ax1.hist(sample_means_20, bins = 20)
ax2.hist(sample_means_100, bins = 20)
ax3.hist(sample_means_500, bins = 20)

ax1.set_xlim([0.3 , 0.7])
ax2.set_xlim([0.3 , 0.7])
ax3.set_xlim([0.3 , 0.7])

plt.suptitle('Sample Means for different number of samples drawn_
             ↪[Uniform(0,1)]')
ax1.set_title('n = 20')
```

```
ax2.set_title('n = 100')
ax3.set_title('n = 500')

plt.show()
```



We can conclude from the above plots that as the number of samples drawn ( $n$ ) increases, the probability of obtaining the mean closer to the parameter mean of 0.5 increases. From the histograms, as  $n \rightarrow \infty$ ,  $P(\bar{X}_n = \mu) \rightarrow 1$ . This proves that sample mean is a consistent estimator.