yz709-DS-sup3

Ex3.1
Ex3.4
Ex3.6
Ex4.2
Ex4.3
Ex4.4
Ex4.7
Ex4.9

Ex3.1

Question 1. We are given a dataset $x_1, \ldots, x_n$ which we believe is drawn from Normal(μ, σ^2) where μ and σ are unknown.

(a) Find the maximum likelihood estimators $\hat{\mu}$ and $\hat{\sigma}$.

(b) Find a 95% confidence interval for $\hat{\sigma}$, using parametric resampling.

(c) Repeat, but using non-parametric resampling.

Ex3.1 (a)

$$X \sim Norm(\mu, \sigma^2)$$

$$Pr(X=x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$Pr(X = x_1, x_2, \ldots, x_n) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\log Pr(X = x_1, x_2, \ldots, x_n) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{d \log Pr(X = x_1, x_2, \ldots, x_n)}{d\mu} = 0 \qquad \frac{d \log Pr(X = x_1, x_2, \ldots, x_n)}{d\sigma} = 0$$

$$\frac{1}{2\sigma^2} \sum_{i=1}^{n} 2(x_i - \mu) = 0 \qquad -\frac{n}{2} \times \frac{1}{2\pi\sigma^2} \times 4\pi\sigma + \left[\sum_{i=1}^{n}(x_i-\mu)^2\right] \cdot \frac{1}{\sigma^3} = 0$$

$$\sum_{i=1}^{n} x_i = n\mu \qquad -\frac{n}{\sigma} + \sum_{i=1}^{n}(x_i-\mu)^2 \cdot \frac{1}{\sigma^3} = 0$$

$$\Rightarrow \hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x} \qquad \because \hat{\mu} = \bar{x}$$

$$\therefore \sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

$$\Rightarrow \hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

- (b):

```
x = [x1,x2,...,xn]
n = len(x)

# 1. define the readout statistic
def mu_hat(x): return np.mean(x)
def sigma_hat(x): return np.sqrt(np.mean((x-mu_hat(x))**2))

# 2. generate a synthetic dataset
def rx(): return np.random.normal(loc=mu_hat(x), scale=sigma_hat, size=n)

# 3. sample the readout statistic and report its spread
sigma_hat_ = [sigma_hat(rx()) for i in range(10000)]
lo,hi = np.quantile(mu_hat_, [.025, .975])
print(f"95% confidence interval [{lo:.3},{hi:.3}]")
```

- (c):

```
# non-parametric resampling
x = [x1,x2,...,xn]
```

```
# 1. define the readout statistic
def mu_hat(x): return np.mean(x)
def sigma_hat(x): return np.sqrt(np.mean((x-mu_hat(x))**2))

# 2. use the empirical distribution of the dataset, pick values at random
def rx(): return np.random.choice(x, size=len(x))

# 3. sample the readout statistic and report its spread
sigma_hat_ = [sigma_hat(rx()) for _ in range(10000)]
lo, hi = np.quantile(sigma_hat_, [0.025, 0.975])
print(f"95% confidence interval [{lo:.3},{hi:.3}]")
```

💡 Comments:

Ex3.4

Question 4. I toss a coin n times and get the answers $x_1, \ldots, x_n$. My model is that each toss is $X_i \sim \text{Bin}(1, \theta)$, and I wish to test the null hypothesis that $\theta \geq 1/2$.

(a) Find an expression for $\Pr(x_1, \ldots, x_n\,;\,\theta)$. Give your expression as a function of $y = \sum_i x_i$.

(b) Sketch $\log \Pr(x_1, \ldots, x_n\,;\,\theta)$ as a function of θ, for two cases: $y < n/2$, and $y > n/2$.

(c) Assuming H_0 is true, what is the maximum likelihood estimator for θ?

(d) Let the test statistic be y. What is the distribution of this test statistic, when θ is equal to your value from part (c)?

(e) Explain why a one-sided hypothesis test is appropriate. Give an expression for the p-value of the test.

Ex3.4

(a) $X \sim Bin(n, \theta)$

$$Pr(x_1, x_2, \dots, x_n; \theta) = \binom{n}{y} \theta^y (1-\theta)^{n-y} \quad \text{where } y = \sum_i x_i$$

$\in [0, 1]$

(b) $\log Pr(x_1, x_2, \dots, x_n; \theta) = \log\binom{n}{y} + y\log\theta + (n-y)\log(1-\theta) \quad < 0$

$$\frac{d\log Pr}{d\theta} = \frac{y}{\theta} + \frac{y-n}{1-\theta} = 0 \qquad y(1-\theta) + \theta(y-n) = 0$$

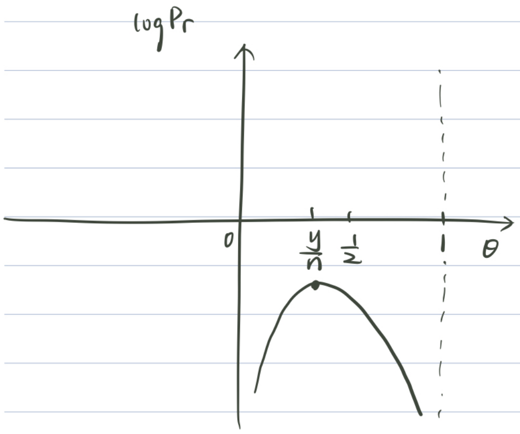
$$y = n\theta$$

$$\theta = \frac{y}{n}$$

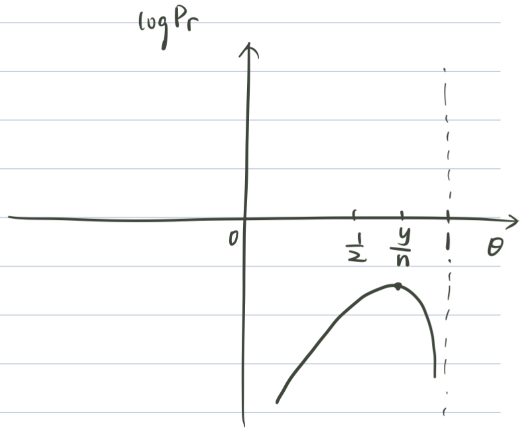
$\theta \to 0^+ \quad \log Pr \to -\infty$

$\theta \to 1^- \quad \log Pr \to -\infty$

when $y < \frac{n}{2}$,



when $y > \frac{n}{2}$,



(c) Assume $\theta \geq \frac{1}{2}$

then $\hat{\theta} = \max(\frac{y}{n}, \frac{1}{2})$

(d) since $\hat{\theta} = \max(\frac{y}{n}, \frac{1}{2}) = \max(\frac{\Sigma X_i}{n}, \frac{1}{2})$

$= \max(\bar{X}, \frac{1}{2})$

y is the sample mean, when we generate 10000 (or a big number) of n coin toss each, we could apply the central limit theorem,

y follows a normal distribution of $Norm(n\theta, \frac{n\theta(1-\theta)}{n})$

$Y \sim Norm(n\theta, \theta(1-\theta))$

(e) $P = Pr_Y(y < \frac{n}{2}) = \int_0^{\frac{n}{2}} \frac{1}{2\pi\theta(1-\theta)} e^{-\frac{(y-n\theta)^2}{2\theta(1-\theta)}} dy$

One-sided hypothesis: because the null hypothesis has a direction about the effect ($\theta \geq \frac{1}{2}$)

hence we are only interested in testing whether $\hat{\theta} = \frac{y}{n} < \frac{1}{2}$

(the MLE of $\hat{\theta}$)

💡 Comments:

Ex3.6

Question 6. A recent paper *Historical language records reveal a surge of cognitive distortions in recent decades* by Bollen et al., `https://www.pnas.org/content/118/30/e2102061118.full`, claims that depression-linked turns of phrase have become more prevalent in recent decades. This paper reports both confidence intervals and null hypotheses. Explain how it is computes them, in particular (1) the readout statistic, (2) the sampling method.

Let the dataset be k n-grams $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$ where $y_i = [y_{i,1855}, ..., y_{i,2020}]$ be a vector giving the prevalence of n-gram i in each year and $x_i \in \{1, 2, 3, 4, 5\}$ be the number of words in that n-gram.

- (1) the readout statistic: $t(x_1, ..., x_k)$ is a vector which represents the yearly z-score values
- (2) the sampling method:

- For the null model: 10000 fold non-parametric sampling (bootstrapping) of 241 randomly chosen n-grams from all n-grams in the combined set of English, Spanish and German Google n-gram corpus such that they have the same number of 1- to 5- gram (English 241 n-grams, Spanish 435 n-grams and German 296 n-grams)
- For English, Spanish and German N-gram models: 10000 fold non-parametric sampling (bootstrapping) of 241, 435, and 296 n-grams for English, Spanish and German, respectively.

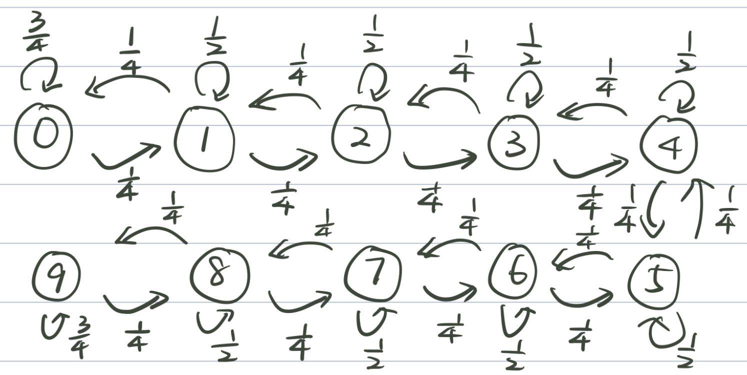
> 💡 Comments:

Ex4.2

Question 2. Draw the state space diagram for this Markov chain.

```
def rw():
    MAX_STATE = 9
    x = 0
    while True:
        yield x
        d = numpy.random.choice([-1,0,1], p=[1/4,1/2,1/4])
        x = min(MAX_STATE, max(0, x + d))
```

Ex 4.2



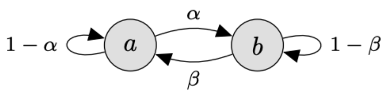
states from 0 to 9

$\xrightarrow{P}$ edge with transition probability

Comments:

Ex4.3

Question 3. Here is the state space diagram for a Markov chain. Find a stationary distribution. Is it unique?



- Yes, the stationary distribution is unique, because the state space diagram for this Markov chain is irreducible, you could get from any state to another state.

$\pi_a = \beta \pi_b + (1-\alpha)\pi_a \quad \Rightarrow \alpha \pi_a = \beta \pi_b$

$\pi_b = \alpha \pi_a + (1-\beta)\pi_b$

$\sum \pi_i = 1$ since π is a probability distribution

$\Rightarrow \pi_\alpha + \pi_b = 1$

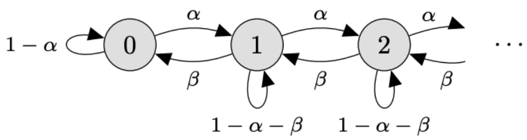
$(\frac{\beta}{\alpha} + 1)\pi_b = 1$

$\pi_b = \frac{\alpha}{\alpha+\beta}$, $\pi_a = \frac{\beta}{\alpha+\beta}$

Comments:

Ex4.4

Question 4. Here is the state space diagram for a Markov chain, with state space $\{0, 1, 2, \ldots\}$. It is parameterized by α and β, with $0 < \alpha < \beta$ and $\alpha + \beta < 1$. Let $\pi_n = (1 - {}^{\alpha}/_{\beta})({}^{\alpha}/_{\beta})^n$, $n \geq 0$. Show that π is a stationary distribution.



Need to verify π_n indeed solve the equations

① Verify $\sum_{n=0}^{\infty} \pi_n = 1$ to show π is a valid distribution

$$\sum_{n=0}^{\infty} \left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^n = \left(1-\frac{\alpha}{\beta}\right) \sum_{n=0}^{\infty} \left(\frac{\alpha}{\beta}\right)^n \quad \text{since } 0<\alpha<\beta.$$

$$= \left(1-\frac{\alpha}{\beta}\right) \frac{1}{\left(1-\frac{\alpha}{\beta}\right)} = 1 \quad \text{as required}$$

② for $x=0$: $\pi_0 = (1-\alpha)\pi_0 + \beta\pi_1$

for $x>0$: $\pi_x = (1-\alpha-\beta)\pi_x + \alpha\pi_{x-1} + \beta\pi_{x+1}$

$(\alpha+\beta)\pi_x = \alpha\pi_{x-1} + \beta\pi_{x+1}$

Assume $\pi_x = \left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^x$,

for π_0 :

$LHS = \left(1-\frac{\alpha}{\beta}\right)$

$RHS = (1-\alpha)\left(1-\frac{\alpha}{\beta}\right) + \beta\left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)$

$= (1-\alpha+\alpha)\left(1-\frac{\alpha}{\beta}\right) = 1-\frac{\alpha}{\beta}$

$\Rightarrow LHS = RHS$

for π_x, $x>0$:

$LHS = (\alpha+\beta)\left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^x$

$RHS = \alpha\left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^{x-1} + \beta\left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^{x+1}$

$= \left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^{x-1}\left(\alpha + \beta\cdot\left(\frac{\alpha}{\beta}\right)^2\right)$

$= \left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^{x-1}\left(\alpha + \frac{\alpha^2}{\beta}\right)$

$= \left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^{x-1} \frac{\alpha(\beta+\alpha)}{\beta}$

$= (\alpha+\beta)\left(1-\frac{\alpha}{\beta}\right)\left(\frac{\alpha}{\beta}\right)^x$

$\Rightarrow LHS = RHS$

Comments:

Ex4.7

Question 7. Let X_n be a mean-reverting random walk,

$$X_{n+1} = \mu + \lambda(X_n - \mu) + N(0, \sigma^2) \quad \text{where} \quad -1 < \lambda < 1.$$

The stationary distribution for this process is a Normal distribution. Find its parameters.

Ex 4.7

The state space is $\mathbb{R}$, uncountable, so use the definition of stationary distribution for continuous random variable X:

A distribution is stationary if $X_0 \sim \pi \Rightarrow X_1 \sim \pi$

since $X_{n+1} = \mu + \lambda(X_n - \mu) + N(0, \sigma^2)$ where $N(0,\sigma^2)$ is the error term

we know $X_{n+1} \sim Norm(\mu + \lambda(X_n - \mu), \sigma^2)$

if $X_t \sim Norm(\mu_t, \sigma_t^2)$ for the stationary distribution

then

$$\mu_t = E(X_t) = E(\mu + \lambda(X_{t-1} - \mu) + N(0,\sigma^2))$$
$$= \mu + \lambda E(X_{t-1}) - \lambda\mu + 0$$
$$= (1-\lambda)\mu + \lambda E(X_{t-1})$$

since $E(X_{t-1}) = E(X_t) = \mu_t$

we have $\mu_t = (1-\lambda)\mu + \lambda \mu_t$

$\Rightarrow (1-\lambda)\mu_t = (1-\lambda)\mu$

since $-1 < \lambda < 1$,

we have $\mu_t = \mu$

$$\sigma_t^2 = V(X_t) = V(\mu + \lambda(X_{t-1} - \mu) + N(0,\sigma^2))$$
$$= \lambda^2 V(X_{t-1}) + \sigma^2$$

since $V(X_{t-1}) = V(X_t) = \sigma_t^2$

we have $\sigma_t^2 = \lambda^2 \sigma_t^2 + \sigma^2$

$$\sigma_t^2 = \frac{\sigma^2}{1-\lambda^2}$$

hence the stationary distribution is $Norm(\mu, \frac{\sigma^2}{1-\lambda^2})$

💡 Comments:

Ex4.9

Question 9. Consider a moving object with noisy location readings. Let X_n be the location at timestep $n \geq 0$, and Y_n the reading. Here's the simulator.

```
def hmm():
    MAX_STATE = 9
    x = numpy.random.randint(low=0, high=MAX_STATE+1)  # initial location X₀
    while True:
        e = numpy.random.choice([-1,0,1])
        y = min(MAX_STATE, max(0, x + e))  # noisy reading of location
        yield y
        d = numpy.random.choice([-1,0,1], p=[1/4,1/2,1/4])
        x = min(MAX_STATE, max(0, x + d))  # new location at next timestep
```

We'd like to infer the location X_n, given readings $y_0, \dots, y_n$.

(a) Give justifications for the following three equations, which give an inductive solution. First the base case,

$$\Pr(x_0 \mid y_0) = \text{const} \times \Pr(x_0) \Pr(y_0 \mid x_0),$$

and next two equations for the induction step,

$$\Pr(x_n \mid h) = \sum_{x_{n-1}} \Pr(x_{n-1} \mid h) \Pr(x_n \mid x_{n-1})$$
$$\Pr(x_n \mid h, y_n) = \text{const} \times \Pr(x_n \mid h) \Pr(y_n \mid x_n).$$

- Base case: use Bayes's rule where $Pr(y_0)$ is a constant

$$Pr(x_0|y_0) = \frac{Pr(y_0|x_0)Pr(x_0)}{Pr(y_0)} = const \cdot Pr(x_0)Pr(y_0|x_0)$$

- Inductive step 1: use the law of total probability and memorylessness, and all applied on conditional probability scenario

$$\begin{aligned} Pr(x_n|h) &= \sum_{x_{n-1}} Pr(x_n|x_{n-1}, h)Pr(x_{n-1}|h) \\ &= \sum_{x_{n-1}} Pr(x_{n-1}|h)Pr(x_n|x_{n-1}) \end{aligned}$$

- Inductive step 2: use Bayes's rule for conditional probability scenario, and y_n only depends on x_n

$$\begin{aligned} Pr(x_n|h, y_n) &= \frac{Pr(y_n|x_n, h)Pr(x_n|h)}{Pr(y_n|h)} \\ &= const \cdot Pr(y_n|x_n)Pr(x_n|h) \end{aligned}$$

(b) Give pseudocode for a function that takes as input a list of readings $(y_0, \dots, y_n)$ and outputs the probability vector

$$[\pi_0, \dots, \pi_{\mathsf{MAX\_STATE}}], \qquad \pi_x = \mathbb{P}(X_n = x \mid y_0, \dots, y_n).$$

$$Pr(X_n = x|y_0, y_1, ..., y_n) = const \cdot Pr(x_n|y_0, y_1, ..., y_{n-1}) \cdot Pr(y_n|x_n)$$
$$= const \cdot (\sum_{x_{n-1}} Pr(x_{n-1}|y_0, y_1, ..., y_{n-1}) Pr(x_n|x_{n-1})) \cdot Pr(y_n|x_n)$$
$$\text{let } P_{ij} = \mathbb{P}(X_{n+1} = j|X_n = i) \text{ be the transition matrix}$$
$$\text{let } Q_{ij} = \mathbb{P}(Y_n = j|X_n = i) \text{ be the emission matrix}$$
$$\text{let } \pi^{(n)} \text{ be the probability vector at time n}$$

```
x = [0,1,2,3,4,5,6,7,8,9] # initial probability = 1/10
y = [y0,...,yn]

# calculate π at time 0
π[0] = [(1/10)*Q[i][y0] for i in range(10)]
π[0] = π[0] / np.sum(π[0])

for j in range(1,n+1):
  # induction 1:
  π[j] = [np.sum([π[j-1][k] * P[k][i] for k in range(10)])
             for i in range(10)]

  # induction 2:
  π[j] = [π[j][i] * Q[i][yj] for i in range(10)]
  π[j] =  π[j] / np.sum(π[j])
```

(c) If your code is given the input $(3, 3, 4, 9)$, it should fail with a divide-by-zero error. Give an interpretation of this failure.

- The transition from $y_2 = 4$ and $y_3 = 9$ is not possible because the transition from hidden state $x_4 = 3, 4, or\ 5$ to the hidden state $x_5 = 8\ or\ 9$ is impossible.
- Hence when `j = 3`, `Q[i][y3] = 0` when $i \in [0, 7]$(inclusive), `np.sum([π[j-1][k] * P[k][i]) = 0` when $i \in [8, 9]$(inclusive) and $k \in [0, 9]$(inclusive)
- Therefore, `π[j] =  π[j] / np.sum(π[j])` gives a divide-by-zero error

💡 Comments: