Applied Statistics (ECS764P) - Lab 2

Fredrik Dahlqvist

3 Nov 2022

1 Theory

1. Normal distributions have the following two properties:

   - the sum of two normals is normal: $N(\mu_1, \sigma_1) + N(\mu_2, \sigma_2) = N\left(\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2}\right)$
   - re-scaling a normal gives a normal: for any $\alpha > 0$, $\alpha N(\mu, \sigma) = N(\alpha\mu, \alpha\sigma)$

   Use these two facts to compute the distribution of sample means for identically and normally distributed independent samples of length n. Specifically, compute the distribution of

   $$\frac{1}{n}\sum_{i=1}^{n} N(\mu, \sigma)$$

2. Consider the family of distributions

   $$\text{Poisson}(\lambda), \lambda \in \mathbb{R}$$

   Show that the MLE $\hat{\lambda}$ is given by the sample mean.

3. Using the definition of the sum of two probability measures given during the lectures, show that the sum of two identical and independent Bernoulli distributions $\text{Bern}(p)$ is given by a binomial distribution $\text{Binom}(2, p)$. Formally show that

   $$\text{Bern}(p) + \text{Bern}(p) = \text{Binom}(2, p)$$

   (Hint: What is the support of $\text{Bern}(p) + \text{Bern}(p)$*? What is the support of* $\text{Binom}(2, p)$*? Do the two probability measures agree on every element of their support? If yes, then they are equal.)*

2 Practice

1. Import `scipy.stats` in order to access the `scipy.stats.beta` distribution. Using the cdf method of `scipy.stats.beta` define a function called `beta_measure` which will take two arguments `a, b` and which will return the probability mass of the interval $[a, b]$ under the probability measure $\text{Beta}(3, 7)$, i.e.

   $$\text{Beta}(3, 7)([a, b])$$

   Test your function by printing the result of:

   (a) `beta_measure(0,1)`

   (b) `beta_measure(0,0)`

   (c) `beta_measure(0.25,0.75)`

(d) `beta_measure(0,0.5)`

(e) `beta_measure(0.5,1)`

Plot the pdf of Beta $(3, 7)$. Do the probability masses you've printed tally with the shape of the density?

2. Using the pdf method of `scipy.stats.beta` define a function called `beta_pdf` which will take one argument `x` and return the pdf of the probability measure Beta $(3, 7)$ evaluated at `x`. Import the integration routine `quad` from `scipy.integrate`, and have a look at the documentation `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html` to see how it works. Use `quad` to compute and print the following integrals

   (a) $\int_0^1 \texttt{beta\_pdf}(x)\ dx$

   (b) $\int_0^0 \texttt{beta\_pdf}(x)\ dx$

   (c) $\int_{1/4}^{3/4} \texttt{beta\_pdf}(x)\ dx$

   (d) $\int_0^{1/2} \texttt{beta\_pdf}(x)\ dx$

   (e) $\int_{1/2} \texttt{beta\_pdf}^1(x)\ dx$

   Compare your answers with those of the previous question.

3. Recall from the lectures that if a probability distribution d_1 has density f_1 and a probability distribution d_2 has density f_2, then the density of the sum $d_1 + d_2$ is given by the convolution of the two densities, viz.

$$f_{1+2}(t) = \int_{\infty}^{\infty} f_1(x) f_2(t - x)\ dx.$$

   In this question we consider the sum of Beta $(3, 7)$ + Beta $(7, 3)$. What is the support of Beta $(3, 7)$? What is the support of Beta $(7, 3)$? Therefore, what is the support of Beta $(3, 7)$ + Beta $(7, 3)$?

   Write a function which implements the integrand of the integral above, that is to say that implements $f_1(x)f_2(t-x)$, where f_1 is the density of Beta $(3, 7)$ and f_2 is the density of Beta $(7, 3)$. (*Hint: this function will need two arguments.*)

   Next, generate 100 points $(t_1, \ldots, t_{100})$ along the support of Beta $(3, 7)$ + Beta $(7, 3)$ (using `numpy`'s `linspace` function), and using a `for` loop, compute the pdf $f_{1+2}(t_i)$ at these 100 points using `quad`. (*Hint: the documentation of* `quad` *has an example showing how to integrate a function with two arguments along its first argument.*) Plot your result.

   Finally, generate 10000 samples from Beta $(3, 7)$, 10000 samples from Beta $(7, 3)$ *independently*, add them, and plot the histogram of these sums along with the pdf computed in the previous step. What do you observe?

4. Install `pandas-datareader` (do `pip install pandas-datareader` in your terminal). With this library it is very easy to download data from Yahoo Finance (and other providers too). Download the last 10 years of Microsoft stock using

```
my_data = data.DataReader('MSFT', 'yahoo', '2012-11-02', '2022-11-02')
```

   Keep the "Close" column and use it to compute the time series of (percentage) *daily returns* using the formula

$$\text{Return}_t = 100 \times \left(\frac{\text{Close}_t}{\text{Close}_{t-1}} - 1 \right)$$

   Warning: do **not** make a local copy of this data! It is easier, cleaner and less error-prone to access it directly from Yahoo Finance using `pandas-datareader`.

   Plot the histogram of daily returns. Find a family of distributions which you think would model this distribution well. (*Hint: what is the support of the daily returns? Is it symmetric or skewed? Has it got fat tails/positive excess kurtosis?).*

   The continuous distributions in `scipy.stats` have a method called `fit` which, given some data, computes the Maximum Likelihood Estimators for the parameters of the distribution. Use this method to find the optimal probability distribution in the family you have chosen, and plot the corresponding pdf alongside the histogram of observed daily returns.

   Finally, plot the QQ plot of the daily returns data versus the model you have just fitted. Comment on the quality of your fit.