Computational Statistics and Data Analysis (MVComp2)

Solutions to exercise 2

Lecturer Tristan Bereau

Semester Wi23/24
Due Nov. 2, 2023, 23:59

1 Coin-tossing game (2 points)

You play a game that consists of tossing two coins. You win €1 if both coins land on tails, you win €2 if both coins land on heads, and lose €1 otherwise.

(a) Calculate the mean and variance of your winnings on a single play of the game.

(b) What is the fair price to play this game (i.e., payoff and cost of playing have mean 0)?

1.1 Solution

There are four outcomes to this game:

Coin 1	Coin 2	Payout
H	H	+€2
T	T	+€1
H	T	−€1
T	H	−€1

(a)

Mean $\mu = \frac{1}{4}(2 + 1 - 1 - 1) = 0.25$
Variance $\mathrm{Var}[X] = \frac{1}{4}[(2-\mu)^2 + (1-\mu)^2 + (-1-\mu)^2 + (-1-\mu)^2] = 1.6875$

(b) The price to play should equate the odds of winning: +€0.25.

2 Expectations and variances (3 points)

Let X, Y be discrete random variable and a, b be constants. Prove the following relations:

(a) $\mathrm{Var}[aX + b] = a^2\mathrm{Var}[X]$

(b) $E[X] = E_Y[E_X[X|Y]]$

(c) $\mathrm{Var}[X] = E_Y[\mathrm{Var}[X|Y]] + \mathrm{Var}[E_X[X|Y]]$

2.1 Solution

(a) First consider the first moment: $E[aX+b] = a\mu + b$. This leads to

$$\begin{aligned}\mathrm{Var}[aX+b] &= E[(aX+b-(a\mu+b))^2]\\ &= E[(aX+b-a\mu-b)^2]\\ &= a^2E[(X-\mu)^2]\\ &= a^2\mathrm{Var}[X]\end{aligned}$$

(b)

$$\begin{aligned}E_Y[E_X[X|Y]] &= E_Y\left[\sum_x xp(X=x|Y=y)\right]\\ &= \sum_y\sum_x xp(X=x|Y=y)p(Y=y)\\ &= \sum_x xp(X=x)\\ &= E[X]\end{aligned}$$

where we made use of the law of total probability.

(c) Recall the property $\mathrm{Var}[X] = E[X^2] - E[X]^2$.

In addition, from (b) we know that $E[X] = E_Y[E_X[X|Y]]$. Similarly: $E[X^2] = E_Y[E_X[X^2|Y]]$.

$$\begin{aligned}\mathrm{Var}[X] &= E[X^2] - E[X]^2\\ &= E_Y[E_X[X^2|Y]] - (E_Y[E_X[X|Y]])^2\\ &= E_Y[E_X[X^2|Y] - E_X[X|Y]^2] + E_Y[E_X[X|Y]^2] - (E_Y[E_X[X|Y]])^2\\ &= E_Y[\mathrm{Var}[X|Y]] + \mathrm{Var}[E_X[X|Y]]\end{aligned}$$

3 Covariance and correlation (2 points)

Prove that the correlation coefficient, ρ, is bounded by -1 and 1.

3.1 Solution

Recall the definition of the correlation coefficient

$$\rho(X,Y) = \frac{\mathrm{Cov}[X,Y]}{\sigma_X\sigma_Y} = \frac{\mathrm{Cov}[X,Y]}{\sqrt{\mathrm{Var}[X]}\sqrt{\mathrm{Var}[Y]}}.$$

Furthermore, $\mathrm{Cov}[X,Y] = E[(X-\mu_x)(Y-\mu_y)]$ and $\mathrm{Var}[X] = E[(X-\mu_x)^2]$.

From the Cauchy-Schwarz inequality, we obtain

$$|\mathrm{Cov}[X,Y]|^2 \leq \mathrm{Var}[X]\mathrm{Var}[Y]$$

which yields $|\rho| \leq 1$.

4 Correlation Between CO_2 levels and Earth's surface temperature (3 points)

You set out to investigate the correlations between mean CO_2 levels and Earth's surface temperature over the last few decades. Datasets are available:

1. Mean monthly CO_2 levels from the Mauna Loa Observatory dataset, which provides a continuous record from 1958 to the present. CSV file `monthly_in_situ_co2_mlo.csv` available at: https://scrippsco2.ucsd.edu/data/atmospheric_co2/primary_mlo_co2_record.html
2. Global mean surface temperature datasets, available from NASA's Goddard Institute for Space Studies. CSV file of "Global-mean monthly, seasonal, and annual means" available at: https://data.giss.nasa.gov/gistemp/.

Procedure:

- Collect the data for the same time frame.
- Clean the data of any outliers or missing values.
- Calculate annual means for both datasets.

(a) Determine the (Pearson) correlation coefficient between annual CO_2 levels and temperature deviation.

(b) Visualize the correlation using a parity plot (i.e., temperature deviation vs. CO_2 levels.)

Hint: If using Python, you may find the following `pandas` functions useful: `read_csv`, `groupby`, `merge_asof`.

4.1 Solution

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Temperature deviations from 1951-1980 means
df_temp = pd.read_csv(
  "https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.csv",
  delimiter=",",
  skiprows=1,
)

# Monthly average CO2 concentration from Mauna Lau Observatory, Hawaii
url_co2 = (
  "https://scrippsco2.ucsd.edu/assets/data/atmospheric/stations/"
  + "in_situ_co2/monthly/monthly_in_situ_co2_mlo.csv"
)
df_co2 = pd.read_csv(
  url_co2,
  skiprows=60,
).iloc[:,:5]
df_co2.columns = [
  "year", "month", "date", "numeric_year", "co2"
]
```

```python
# Data cleaning
df_temp = df_temp.replace("***", np.nan).dropna()
df_temp["temperature"] = df_temp["J-D"].astype(float)
df_co2 = df_co2.replace(-99.99, np.nan).dropna()
df_co2 = df_co2.groupby("year").sum()
df_co2 = df_co2.loc[df_co2["month"] == 78] # Only keep full years

# Merge the datasets
df = pd.merge_asof(df_co2, df_temp, left_on="year", right_on="Year").dropna()

# Visualize the results
plt.scatter(df["co2"], df["temperature"], c=df["Year"])
plt.xlabel(r"Annual average CO$_2$ concentration [ppm]")
plt.ylabel("Temperature deviation [$^\circ$C]")
plt.colorbar()
plt.grid()
corr_coeff = df[["co2", "J-D"]].corr().iloc[0,1]
plt.title(f"Correlation coefficient: {corr_coeff:.3f}");
```