ExamJanuary_2023_problem

January 19, 2023

1 Exam 5th of January 2023, 8.00-13.00 for the course 1MS041 (Introduction to Data Science / Introduktion till dataanalys)

1.1 Instructions:

1. Complete the problems by following instructions.
2. When done, submit this file with your solutions saved, following the instruction sheet.

This exam has 3 problems for a total of 40 points, to pass you need 20 points.

1.2 Some general hints and information:

- Try to answer all questions even if you are uncertain.
- Comment your code, so that if you get the wrong answer I can understand how you thought this can give you some points even though the code does not run.
- Follow the instruction sheet rigorously.
- This exam is partially autograded, but your code and your free text answers are manually graded anonymously.
- If there are any questions, please ask the exam guards, they will escalate it to me if necessary.
- I (Benny) will visit the exam room at around 10:30 to see if there are any questions.

1.3 Tips for free text answers

- Try to be very clear with your reasoning, there should be zero ambiguity in what you are referring to.
- If you want to include math, you can write LaTeX in the Markdown cells, for instance `$f(x)=x^2$` will be rendered as $f(x) = x^2$ and `$$f(x) = x^2$$` will become an equation line, as follows

  $$f(x) = x^2$$

  Another example is `$$f_{Y \mid X}(y,x) = P(Y = y \mid X = x) = \exp(\alpha \cdot x + \beta)$$` which renders as

  $$f_{Y \mid X}(y,x) = P(Y = y \mid X = x) = \exp(\alpha \cdot x + \beta)$$

1.4 Finally some rules:

- You may not communicate with others during the exam, for example:
  - You cannot ask for help in Stack-Overflow or other such help forums during the Exam.
  - You may not communicate with AI's, for instance ChatGPT.
  - Your on-line and off-line activity is being monitored according to the examination rules.

1.5 Good luck!

[ ]:

```
# Insert your anonymous exam ID as a string in the variable below
examID="XXX"
```

1.6 Exam vB, PROBLEM 1

Maximum Points = 14

A courier company operates a fleet of delivery trucks that make deliveries to different parts of the city. The trucks are equipped with GPS tracking devices that record the location of each truck at regular intervals. The locations are divided into three regions: downtown, the suburbs, and the countryside. The following table shows the probabilities of a truck transitioning between these regions at each time step:

Current region	Probability of transitioning to downtown	Probability of transitioning to the suburbs	Probability of transitioning to the countryside
Downtown	0.3	0.4	0.3
Suburbs	0.2	0.5	0.3
Countryside	0.4	0.3	0.3

1. If a truck is currently in the suburbs, what is the probability that it will be in the downtown region after two time steps? [2p]
2. If a truck is currently in the suburbs, what is the probability that it will be in the downtown region **the first time** after two time steps? [2p]
3. Is this Markov chain irreducible? Explain your answer. [3p]
4. What is the stationary distribution? [3p]
5. Advanced question: What is the expected number of steps until the first time one enters the suburbs region having started in the downtown region. Hint: to get within 1 decimal point, it is enough to compute the probabilities for hitting times below 30. Motivate your answer in detail [4p]. You could also solve this question by simulation, but this gives you a maximum of [2p].

[ ]:

```
# Part 1

# Fill in the answer to part 1 below
problem1_p1 = XXX
```

[ ]:

```
# Part 2

# Fill in the answer to part 2 below
problem1_p2 = XXX
```

[ ]:

```
# Part 3
# Fill in the answer to part 3 below as a boolean
problem1_irreducible = True/False
```

1.7 Part 3

Double click this cell to enter edit mode and write your answer for part 3 below this line.

[ ]:

```
# Part 4

# Fill in the answer to part 4 below
# the answer should be a numpy array of length 3
# make sure that the entries sums to 1!
problem1_stationary = XXX
```

[ ]:

```
# Part 5

# Fill in the answer to part 5 below
# That is, the expected number of steps
problem1_ET = XXX
```

1.8 Part 5

Double click this cell to enter edit mode and write your answer for part 5 below this line.

1.9 Exam vB, PROBLEM 2

Maximum Points = 13

You are given the "Abalone" dataset found in `data/abalone.csv`, which contains physical measurements of abalone (a type of sea shells) and the age of the abalone measured in **rings** (the number of rings in the shell) `https://en.wikipedia.org/wiki/Abalone`. Your task is to train a `linear regression` model to predict the age (Rings) of an abalone based on its physical measurements.

To evaluate your model, you will split the dataset into a training set and a testing set. You will use the training set to train your model, and the testing set to evaluate its performance.

1. Load the data into a pandas dataframe `problem2_df`. Based on the column names, figure out what are the features and the target and fill in the answer in the correct cell below. [2p]
2. Split the data into train and test. [2p]
3. Train the model. [1p]
4. On the test set, evaluate the model by computing the mean absolute error and plot the empirical distribution function of the residual with confidence bands (i.e. using the DKW inequality and 95% confidence). Hint: you can use the function `plotEDF,makeEDF` combo from `Utils.py` that we have used numerous times, which also contains the option to have confidence bands. [3p]
5. Provide a scatter plot where the x-axis corresponds to the predicted value and the y-axis is the true value, do this over the test set. [2p]

6. Reason about the performance, for instance, is the value of the mean absolute error good/bad and what do you think about the scatter plot in point 5? [3p]

[ ]:

```
# Part 1
# Let problem2_df be the pandas dataframe that contains the data from the file
# data/abalone.csv
problem2_df = XXX
```

[ ]:

```
# Part 1

# Fill in the features as a list of strings of the names of the columns

problem2_features = ["XXX"]

# Fill in the target as a string with the correct column name

problem2_target = "XXX"
```

[ ]:

```
# Part 2



# Split the data into train and test using train_test_split
# keep the train size as 0.8 and use random_state=42
problem2_X_train,problem2_X_test,problem2_y_train,problem2_y_test = XXX
```

[ ]:

```
# Part 3

# Include the necessary imports

# Initialize your linear regression model
problem2_model = XXX

# Train your model on the training data
```

[ ]:

```
# Part 4

# Evaluate the model by computing the mean absolute error
problem2_mae = XXX
```

[ ]:

```
# Part 4

# Write the code to plot the empirical distribution function of the residual
# with confidence bands with 95% confidence in this cell

# from Utils import makeEDF,plotEDF
```

[ ]:

```
# Part 5

# Write the code below to produce the scatter plot for part 5
```

1.10 Part 6

Double click this cell to enter edit mode and write your answer for part 6 below this line.

Discussion on the value of the MAE

Discussion on the predicted vs. true scatterplot

Discussion

1.11 Exam vB, PROBLEM 3

Maximum Points = 13

A healthcare organization is interested in understanding the relationship between the number of visits to the doctors office and certain patient characteristics. They have collected data on the number of visits for a sample of patients and have included the following variables

- ofp : number of physician office visits
- ofnp : number of nonphysician office visits
- opp : number of physician outpatient visits
- opnp : number of nonphysician outpatient visits
- emr : number of emergency room visits
- hosp : number of hospitalizations
- exclhlth : the person is of excellent health (self-perceived)
- poorhealth : the person is of poor health (self-perceived)
- numchron : number of chronic conditions
- adldiff : the person has a condition that limits activities of daily living ?
- noreast : the person is from the north east region
- midwest : the person is from the midwest region
- west : the person is from the west region
- age : age in years (divided by 10)
- male : is the person male ?
- married : is the person married ?
- school : number of years of education
- faminc : family income in 10000$
- employed : is the person employed ?
- privins : is the person covered by private health insurance?
- medicaid : is the person covered by medicaid ?

Decide which patient features are resonable to use to predict the target "number of physician office visits". Hint: should we really use the "ofnp" etc variables?

Since the target variable is counts, a reasonable loss function is to consider the target variable as Poisson distributed where the parameter follows $\lambda = \exp(\alpha \cdot x + \beta)$ where α is a vector (slope) and β is a number (intercept). That is, the parameter is the exponential of a linear function. The reason we chose this as our parameter, is that it is always positive which is when the Poisson distribution is defined. To be specific we make the following assumption about our conditional density of $Y \mid X$,

$$f_{Y|X}(y, x) = \frac{\lambda^y e^{-\lambda}}{y!}, \quad \lambda(x) = \exp(\alpha \cdot x + \beta).$$

Recall from the lecture notes, (4.2) that in this case we should consider the log-loss (entropy) and that according to (4.2.1 Maximum Likelihood and regression) we can consider the conditional log-likelihood. Follow the steps of Example 1 and Example 2 in section (4.2) to derive the loss that needs to be minimized.

Hint: when taking the log of the conditional density you will find that the term that contains the $y!$ does not depend on λ and as such does not depend on α, β, it can thus be discarded. This will be essential due to numerical issues with factorials.

Instructions:

1. Load the file `data/visits_clean.csv` into the pandas dataframe `problem3_df`. Decide what should be features and target, give motivations for your choices. [3p]
2. Create the `problem3_X` and the `problem3_y` as numpy arrays with `problem3_X` being the features and `problem3_y` being the target. Do the standard train-test split with 80% training data and 20% testing data. Store these in the variables defined in the cells. [3p]
3. Implement *loss* inside the class `PoissonRegression` by writing down the loss to be minimized, I have provided a formula for the λ that you can use. [2p]
4. Now use the `PoissonRegression` class to train a Poisson regression model on the training data. [2p]
5. Come up with a reasonable metric to evaluate your model on the test data, compute it and write down a justification of this. Also, interpret your result and compare it to something naive. [3p]

[ ]:

```python
# Part 1

# Let problem3_df be the pandas dataframe that contains the data from the file
# data/visits_clean.csv
problem3_df = XXX
```

[ ]:

```python
# Part 1

# Fill in the features as a list of strings of the names of the columns

problem3_features = ["XXX"]

# Fill in the target as a string with the correct column name

problem3_target = "XXX"
```

1.12 Part 1

Double click this cell to enter edit mode and write your answer for part 1 below this line.

What features are reasonable?

In regards to how much data we have, how many features do you think we should aim for?

What other features would you like to have used but was not collected?

Discussion

[ ]:

```
# Part 2

# Fill in your X and y below
problem3_X = XXX
problem3_y = XXX

# Split the data into train and test using train_test_split
# keep the train size as 0.8 and use random_state=42
problem3_X_train, problem3_X_test, problem3_y_train, problem3_y_test = XXX
```

[ ]:

```
# Part 3

# Fill in the function loss below

class PoissonRegression(object):
    def __init__(self):
        self.coeffs = None
        self.result = None

    def fit(self,X,Y):
        import numpy as np
        from scipy import optimize

        # define the objective/cost/loss function we want to minimise
        def loss(coeffs):
            # The parameter lambda for the given X and the proposed values
            # of the coefficients, here coeff[:-1] represent alpha
            # and coeff[-1] represent beta
            lam = np.exp(np.dot(X,coeffs[:-1])+coeffs[-1])

            # use the Y variable that is available here to define
            # the loss function, return the value of the loss for
            # this Y and for this parameter lam defined above
            return XXX
        #Use the loss above together with an optimization method from scipy
        #to find the coefficients of the model
        #this is prepared for you below

        initial_arguments = np.zeros(shape=X.shape[1]+1) # initial guess as 0
        self.result = optimize.minimize(loss, initial_arguments,method='cg')
        self.coeffs = self.result.x

    def predict(self,X):
        #Use the trained model to predict Y
        if (self.coeffs is not None):
            return np.exp(np.dot(X,self.coeffs[:-1])+self.coeffs[-1])
```

[ ]:

```
# Part 4

# Initialize your PoissonRegression model
problem3_model = XXX

# Fit your initialized model on the training data


# This is to make sure that everything went well,
# check that success is True
print(problem3_model.result)
```

[ ]:

```
# Part 5

# Put the computed metric value in the variable below
problem3_metric = XXX
```

1.13 Part 5

Double click this cell to enter edit mode and write your answer for part 5 below this line.

Discussion on reasonable metrics and discussion about the value of the metric

Comparison with a naive model