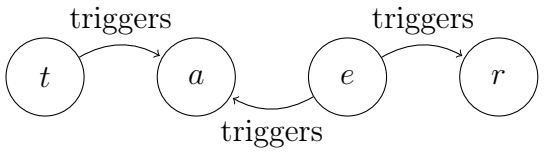
MSAI Statistics Home Assignment 5

Problem 1. (4 points) Consider the following situation. You have an anti-theft alarm a installed in your apartment. It is a good alarm, and it is triggered when a thief t breaks into your apartment. However it may also be triggered if an earthquake e happens. Finally, earthquakes are sometimes announced on the radio r.



We may write the following probabilistic model:

$$p(t, e, a, r) = p(a|t, e)p(r|e)p(t)p(e)$$

Define the following probabilities:

$p(a = 1\|t, e)$	$t = 0$	$t = 1$
$e = 0$	0	1
$e = 1$	0.1	1

	$e = 0$	$e = 1$
$p(r = 1\|e)$	0	0.5

And also define the probabilities $p(t = 1) = 2 \cdot 10^{-4}$ and $p(e = 1) = 10^{-2}$. Now compute the following:

- (2 points) Probability that there is a thief in your apartment if there is an alarm: $p(t = 1|a = 1)$
- (2 points) Probability that there is a thief in your apartment if there is an alarm and you hear an announcement about an earthquake on the radio: $p(t = 1|a = 1, r = 1)$

Problem 2. (2 points) Let $X_1, \ldots, X_n \sim \mathcal{N}(\theta, \sigma^2)$, where σ^2 is known. Consider a prior $\mathcal{N}(a, b^2)$. Show that the posterior is $\mathcal{N}(\bar{\theta}, \tau^2)$, where

$$\bar{\theta} = \frac{1}{\frac{1}{b^2} + \frac{n}{\sigma^2}} \left(\frac{a}{b^2} + \frac{\sum_{i=1}^n X_i}{\sigma^2} \right),$$
$$\tau^2 = \left(\frac{1}{b^2} + \frac{n}{\sigma^2} \right)^{-1}$$

Problem 3. (1 point) Remember Spearman's rank correlation coefficient, which can be written as:

$$\rho_S = \frac{12}{n^3 - n} \sum_{i=1}^{n} \left(i - \frac{n+1}{2} \right) \left(T_i - \frac{n+1}{2} \right)$$

where (R_i, S_i) are the original ranks and (i, T_i) are the ranks sorted by first component.

Prove that we can rewrite this as:

$$\rho_S = 1 - \frac{6}{n^3 - n} \sum_{i=1}^{n} (i - T_i)^2 = 1 - \frac{6}{n(n-1)(n+1)} \sum_{i=1}^{n} (R_i - S_i)^2$$

Hint: expand the series

$$\sum \left[\left(i - \frac{n+1}{2} \right) - \left(T_i - \frac{n+1}{2} \right) \right]^2$$

Problem 4. (4 points) Computer experiment. Use the following code to load the data and get acquainted with it.

```
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
data = load_diabetes(as_frame=True)
print(data["DESCR"])
df = data["frame"]
df_train, df_test = train_test_split(df, test_size=0.2)
```

We will do feature selection on **train dataset** in three ways:

- (1 point) Test independence hypothesis for every feature with target (10 tests total). Remember the normality assumption! Don't forget to account for multiple testing! Fit a linear regression model using features for which we reject the independence hypothesis. Measure the error (anything you like, e.g. RMSE or R^2) on testing dataset.
- (1 point) Train a regularized regression model with all features considered. Remember the normality assumption! Read the summary of your fit. Find the confidence intervals for every coefficient. Fit a new ordinary linear regression model excluding all features that have zero in the confidence interval. Measure the error (same as before) on testing dataset.
- (1 point) Train a linear regression model for every possible subset of features (2^{10} models) and select the best model using Akaike information criteria. To use AIC you will need a probabilistic model, which are available in `statsmodels` (please **don't** use `LassoLarsIC` from `sklearn.linear_model`):

  ```
  import statsmodels.api as sm
  model = sm.OLS(targets, inputs)
  result = model.fit()
  aic = result.aic
  ```

  Measure the error (same as before) on testing dataset.
- (1 point) Compare feature sets and test errors of models from two previous steps.

Problem 5. (5 bonus points) Let $X_1, \ldots, X_n \sim Poisson(\lambda)$.

- (2 bonus points) Consider a prior $\lambda \sim Gamma(\alpha, \beta)$. Show that the posterior is also a Gamma, find its parameters.
- (1 bonus point) Find the posterior mean, show that it is a weighted sum of MLE and prior mean.
- (2 bonus points) Find the Jeffreys' prior, find parameters of the posterior if the prior is Jeffreys' prior.

Problem 6. (3 bonus points) Computer experiment. Use the following code to load the data and get acquainted with it.

```
from sklearn.datasets import load_wine
data = load_wine(as_frame=True)
df = data["frame"]
colors = df["color_intensity"]
hues = df["hue"]
```

Compute the Pearson correlation coefficient between `colors` and `hues`. Remember normality assumption! Provide estimates, tests, and confidence intervals.