Aniket Sanjay Gujarathi
1007629979

Assignment 1

CSC 2506
February 4, 2023

Q1. Decision Theory

One successful use of probabilistic models is for building spam filters, which take in an email and take different actions depending on the likelihood that it's spam.

Imagine you are running an email service. You have a well-calibrated spam classifier that tells you the probability that a particular email is spam: $p(spam|email)$. You have four options for what to do with each email: You can list it as important email, show it to the user, put it in the spam folder, or delete it entirely.

Depending on whether or not the email really is spam, the user will suffer a different amount of wasted time for the different actions we can take, $L(action, spam)$:

Action	Spam	Not spam
Important	15	0
Show	5	1
Folder	1	40
Delete	0	150

Figure 1: Loss Table

Q1.1

Plot the expected wasted user time for each of the three possible actions, as a function of the probability of spam: $p(spam|email)$.

```
import numpy as np
import matplotlib.pyplot as plt

losses = [[15, 0],[5, 1], [1, 40],[0, 150]]
actions_names = ['Important','Show', 'Folder', 'Delete']
num_actions = len(losses)
def expected_loss_of_action(prob_spam, action):
    #TODO: Return expected loss over a Bernoulli random variable
    # with mean prob_spam.
    # Losses are given by the table above.
    total_expected_loss = []
    for prob in prob_spam:
      expected_loss = prob * losses[action][0] + (1 - prob) * (losses[action][1])
      total_expected_loss.append(expected_loss)

    return total_expected_loss

prob_range = np.linspace(0., 1., num=600)
```

```
# Make plot
for action in range(num_actions):
    plt.plot(prob_range, expected_loss_of_action(prob_range, action), label=actions_names[a

plt.xlabel('$p(spam|email)$')
plt.ylabel('Expected loss of action')
plt.legend()
```

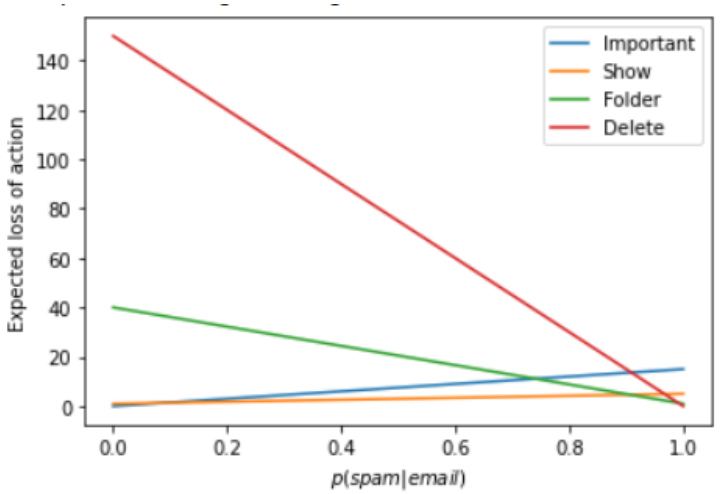

Figure 2: Plot Q1.1

Q1.2

Write a function that computes the optimal action given the probability of spam.

```
def optimal_action(prob_spam):
    #TODO: return best action given the probability of spam.
    #Hint: np.argmin might be helpful.

    # iterate over the prob_spam and find the action that gives least expected error
    optimal_actions = []
    for prob in prob_spam:
      action_losses = []
      for action in range(num_actions):
        expected_loss_action = prob * losses[action][0] + (1 - prob) * (losses[action][1])
        action_losses.append(expected_loss_action)
      optimal_actions.append(np.argmin(action_losses))

    return optimal_actions
```

Q1.3

Plot the expected loss of the optimal action as a function of the probability of spam.

Color the line according to the optimal action for that probability of spam.

```
prob_range = np.linspace(0., 1., num=600)
optimal_losses = {action:[] for action in range(num_actions)}
optimal_actions = []
# TODO: Compute the optimal action and its expected loss for
# probability of spam given by p.
optimal_actions = optimal_action(prob_range)
prob_ranges = {action:[] for action in range(num_actions)}
for i in range(len(prob_range)):
  optimal_losses[optimal_actions[i]].append(prob_range[i] * losses[optimal_actions[i]][0] +
  prob_ranges[optimal_actions[i]].append(prob_range[i])

for action in range(num_actions):
    plt.plot(prob_ranges[action], optimal_losses[action], label=actions_names[action])
plt.xlabel('p(spam|email)')
plt.ylabel('Expected loss of optimal action')
plt.legend()
```

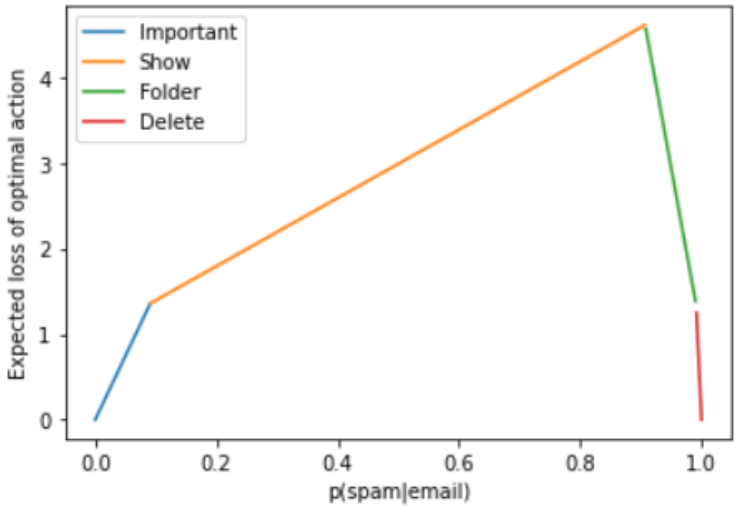

Figure 3: Plot Q1.3

Q1.4

For exactly which range of the probabilities of an email being spam should we delete an email?

Approach: Find the intersection of lines in plot 2 and return the range for which the expected

loss to delete is the lowest.

$$\begin{aligned} important &: y = 15x \\ show &: y = 4x + 1 \\ folder &: y = -39x + 40 \\ delete &: y = -150x + 150 \end{aligned}$$

The intersection of lines with respect to delete line are:

$$\begin{aligned} important &: x = 0.9090, y = 13.6363 \\ show &: x = 0.9675, y = 4.87 \\ folder &: x = 0.9909, y = 1.35135 \end{aligned}$$

As we can see, the expected loss to delete is the lowest at the intersection of the lines delete and folder. The range of probabilities of an email being spam for which we should delete is $0.9909 - 1.0$