CPSC 440/540 Machine Learning – Sample Assignments

1 Hypothesis Testing

We have a dataset generated from two Gaussian distributions with different means but the same variances.

1.1 Neyman-Pearson

Assume $p_0(x) = p(X|H=0)$ and $p_1(x) = p(X|H=1)$ are given by two hypothesis that

$$H_0 : X \sim \mathcal{N}(\mu_0, \sigma^2)$$

$$H_1 : X \sim \mathcal{N}(\mu_1, \sigma^2)$$

where $\mu_0 < \mu_1$

- First, construct a general form of Neyman-Pearson detector.

  Hint: Use log likelihood ratio test with Neyman-Pearson hypothesis testing to build the detector.

  Answer:

  We have a general form of Neyman-Pearson detector based on log likelihood ratio test

  $$\delta(x) = \begin{cases} 1, \ if \ \log(\ell(x)) \geq \eta \\ 0, \ if \ \log(\ell(x)) < \eta \end{cases}$$

  Now perform LLRT

  $$\begin{aligned} \log(\ell(x)) &= \log(\frac{p_1(x)}{p_0(x)}) \\ &= \frac{-(x-\mu_1)^2 + (x-\mu_0)^2}{2\sigma^2} \end{aligned}$$

  If $\log(\ell(x)) \geq \eta$, we have

  $$(x-\mu_0)^2 - (x-\mu_1)^2 \geq 2\sigma^2\eta$$

  $$-2\mu_0 x + \mu_0^2 + 2\mu_1 x - \mu_1^2 \geq 2\sigma^2\eta$$

  $$x \geq \frac{2\sigma^2\eta + \mu_1^2 - \mu_0^2}{2\mu_1 - 2\mu_0}$$

  The general form of Neyman-Person detector is

  $$\delta(x) = \begin{cases} 1, \ if \ x \geq \frac{2\sigma^2\eta + \mu_1^2 - \mu_0^2}{2\mu_1 - 2\mu_0} \\ 0, \ if \ x < \frac{2\sigma^2\eta + \mu_1^2 - \mu_0^2}{2\mu_1 - 2\mu_0} \end{cases}$$

- What are the threshold x^*, and log likelihood ratio threshold η of the Neyman-Pearson test given p-value is 0.05?

  Hint: What is the false alarm rate when knowing p-value?

Answer:

$$\begin{aligned} P_F(\delta) &= \int_{\delta(x)=1} p(x|H=0)dx \\ &= \int_{x^*}^{\infty} p(x|H=0)dx \\ &= 1 - \Phi(\frac{x^* - \mu_0}{\sigma}) \end{aligned}$$

$$\begin{aligned} x^* &= \sigma\Phi^{-1}(1 - P_F(\delta)) + \mu_0 \\ &= \sigma\Phi^{-1}(1 - \alpha) + \mu_0 \\ &= \sigma\Phi^{-1}(0.95) + \mu_0 \\ &= 1.64\sigma + \mu_0 \end{aligned}$$

, where 1.64 is gotten from lookup table.

Rearrange the values in the equation

$$x^* = \frac{2\sigma^2\eta + \mu_1^2 - \mu_0^2}{2\mu_1 - 2\mu_0}$$

We can also get

$$\eta = \frac{(1.64\sigma + \mu_0) * (2\mu_1 - 2\mu_0) - \mu_1^2 + \mu_0^2}{2 * \sigma^2}$$

- Plot the ROC curve multiple Neyman-Pearson detectors. [Optional Open Coding Question]

  Hint: Represent the detection rate with false alarm rate from 0 to 1.

  Answer: Refer to the ROC curve plot in slides.

  Students can specify μ and σ by themselves, and once they are specified, steps for a plot include to include,

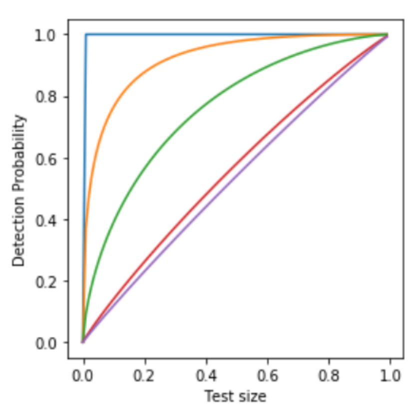
  1) Use threshold x^* to represent P_F;

  2) Use threshold x^* to represent P_D;

  3) Substitute the x^* in P_D formula with P_F;

  4) Given different P_F from 0 - 1 (using np.linspace), get P_D for plotting.

  The plot should be similar to the plot in the textbook Figure 3.1



1.2 Bayesian

Assume we have the same hypothesis that can represent the distributions $p_0(x) = p(X|H=0)$ and $p_1(x) = p(X|H=1)$ provided in last question

$$H_0 : X \sim \mathcal{N}(\mu_0, \sigma^2)$$
$$H_1 : X \sim \mathcal{N}(\mu_1, \sigma^2)$$

where $\mu_0 < \mu_1$

- Construct a bayesian detector given priors and costs are uniform.

  Answer: With uniform priors, this can be degenerated to a MLE problem. The deduction from minimizing risk can be found in slides.

  When $p_1(x) \geq p_0(x)$, we have

$$\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2) \geq \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{x-\mu_0}{\sigma})^2)$$
$$-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2 \geq -\frac{1}{2}(\frac{x-\mu_0}{\sigma})^2$$
$$(\frac{x-\mu_1}{\sigma})^2 \leq (\frac{x-\mu_0}{\sigma})^2$$
$$-2\mu_1 x + \mu_1^2 \leq -2\mu_0 x + \mu_0^2$$
$$x \geq \frac{\mu_1 + \mu_0}{2}$$

  The general form of the detector is

$$\delta(x) = \begin{cases} 1, \ if \ x \geq \frac{\mu_1+\mu_0}{2} \\ 0, \ if \ x < \frac{\mu_1+\mu_0}{2} \end{cases}$$

- Construct a bayesian detector with prior θ for H_1 and uniform costs

  Answer: With uniform costs, this can be degenerated to a MAP problem. The deduction from minimizing risk can be found in slides.

$$\begin{aligned} \delta(x) &= \arg\max_i p(H=i|x) \\ &= \arg\max_i \frac{p(x|H=i)\pi_i}{p(x)} \\ &= \arg\max_i p(x|H=i)\pi_i \end{aligned}$$

  When $p_1(x)\pi_1 \geq p_0(x)\pi_0$, we have

$$\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2)\theta \geq \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{x-\mu_0}{\sigma})^2)(1-\theta)$$
$$\exp(-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2 + \frac{1}{2}(\frac{x-\mu_0}{\sigma})^2) \geq \frac{1-\theta}{\theta}$$
$$-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2 + \frac{1}{2}(\frac{x-\mu_0}{\sigma})^2 \geq \log(\frac{1-\theta}{\theta})$$
$$-(x-\mu_1)^2 + (x-\mu_0)^2 \geq 2\sigma^2 \log(\frac{1-\theta}{\theta})$$

$$2\mu_1 x - \mu_1^2 - 2\mu_0 x + \mu_0^2 \geq 2\sigma^2 \log(\frac{1-\theta}{\theta})$$

$$x \geq \frac{2\sigma^2 \log(\frac{1-\theta}{\theta}) + \mu_1^2 - \mu_0^2}{(2\mu_1 - 2\mu_0)}$$

$$x \geq \frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1 + \mu_0}{2}$$

$$\delta(x) = \begin{cases} 1, \; if \; x \geq \frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1+\mu_0}{2} \\ 0, \; if \; x < \frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1+\mu_0}{2} \end{cases}$$

1.3 Compare Detection with Learning

We have a test dataset with 1000 samples in code/test_data.pkl.

- Now we know that $\mu_0 = -1$, $\mu_1 = 1$, $\sigma = 1$. Give the Neyman-Pearson detector with significance level $\alpha = 0.05$, and a uniform cost bayes-optimum detector with prior for H_1, $\theta = 0.6$. Put them into python class NP and Bayes in the provided code. What are their losses on the test data? [1]

  Answer:

```python
class NP:
    def __init__(self, alpha=0.05, mu0=-1, mu1=1, sigma=1):
        self.threshold = None
        self.alpha = alpha
        self.mu0 = mu0
        self.mu1 = mu1
        self.sigma = sigma
        self.fit()

    def fit(self):
        self.threshold = self.sigma * norm.ppf(1 - self.alpha) + self.mu0

    def predict(self, Xtest):
        return Xtest >= self.threshold

class Bayes:
    def __init__(self, theta=0.6, mu0=-1, mu1=1, sigma=1):
        self.threshold = None
        self.theta = theta
        self.mu0 = mu0
        self.mu1 = mu1
        self.sigma = sigma
        self.fit()

    def fit(self):
        self.threshold = self.sigma ** 2 / (self.mu1 - self.mu0) * math.log((1 - self.theta) / self.theta) + (self.mu1 + self.mu0) / 2

    def predict(self, Xtest):
        return Xtest >= self.threshold
```

[1] The solution code provided is the general form, it would be the same as doing the hand calculation. Either one would be fine

Plug the values into

$$\delta(x) = \begin{cases} 1, \ if \ x \geq 1.64\sigma + \mu_0 \\ 0, \ if \ x < 1.64\sigma + \mu_0 \end{cases}$$

$1.64\sigma + \mu_0 = 0.64$

We get the Neyman-Pearson detector

$$\delta(x) = \begin{cases} 1, \ if \ x \geq 0.64 \\ 0, \ if \ x < 0.64 \end{cases}$$

Plug the values into

$$\delta(x) = \begin{cases} 1, \ if \ x \geq \frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1 + \mu_0}{2} \\ 0, \ if \ x < \frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1 + \mu_0}{2} \end{cases}$$

$\frac{\sigma^2}{\mu_1 - \mu_0} \log(\frac{1-\theta}{\theta}) + \frac{\mu_1 + \mu_0}{2} = \frac{1}{2} \log(0.4/0.6) = \frac{1}{2} * (-0.4) = -0.2$

We get the Bayes-optimum detector

$$\delta(x) = \begin{cases} 1, \ if \ x \geq -0.2 \\ 0, \ if \ x < -0.2 \end{cases}$$

Use them to make predictions on the given test data will get:

Neyman-Pearson detector test error: 25.4%

Bayes-Optimum detector test error: 17.3%

- Now we assume that we don't know the probability distribution, but instead have the training data in code/train_data.pkl. Construct a popular machine learning model LDA, using scikit-learn. Attach the code in the file. What is its the accuracy on test data?

  Answer:

```
class LDA:
    def __init__(self, X=None, y=None):
        self.clf = LinearDiscriminantAnalysis()
        if X is not None and y is not None:
            self.fit(X, y)

    def fit(self, X, y):
        if len(X.shape) == 1:
            X = X[:, np.newaxis]
        self.X = X
        self.y = y
        self.clf.fit(self.X, self.y)

    def predict(self, Xtest):
        if len(Xtest.shape) == 1:
            Xtest = Xtest.to_numpy()[:, np.newaxis]
        return self.clf.predict(Xtest)


@handle("classifier")
def detector():
```

```python
    x, y = load_dataset('train_data', "x", "y")
    model = LDA(x.to_numpy(), y.to_numpy())
    print(f"LDA test error: {eval_model(model, 'test_data'):.1%}")
```

LDA test error: 17.3%

- Compare with three models your have just build [Optional Open Discussion Question]

  Answer: It would be valid as long as students can give reasonings on some of the aspects below

  1) While all of them give a binary and discrete prediction, the first two detectors built with hypothesis testing use distributions based on acquisition of domain knowledge, while the last classifier are learned from data;

  2) Discuss on the performance of different dataset: when the priors used are the same as the test data distribution, bayes-optimum gives the same test error as LDA. This means that even when data are not available, using hypothesis testing with good assumption can give a relative good result.

  3) Twick the parameters, or generate more data with the provided code, then compare the models. For example, give a different significance value for Neyman Pearson detector or give a different prior for Bayes-optimum detector, and see the performance.

2 Short Questions

Sample short questions include:

1. What is the difference between binary hypothesis testing and multiple hypothesis testing?

   Answer: Binary hypothesis testing only tests whether one hypothesis is true or not, while multiple hypothesis testing tests multiple hypothesis simultaneously.

2. What is the likelihood ratio test?

   Answer: The likelihood ratio test is used to determine whether the more complex model, which includes additional parameters or variables, provides a significantly better fit to the data than the simpler model.

3. What is the significance level (alpha) in hypothesis testing? How does it relate to the probability of committing a Type I error?

   Answer: It is the false alarm probability, and it would be the same as the Type 1 error.

4. How to convert a general form Neyman-Pearson detector into a general form Bayes-optimum detector?

   Answer: Replace the likelihood ratio (resp. LLR) thresholds η by the ratio of the prior probabilities $\frac{\pi_0}{\pi_1}$ (More in slides).

5. Differentiate between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP) detectors in the context of hypothesis testing.

   Answer: MLE detectors have a uniform prior, while MAP detectors have different priors for different hypothesis.