CX4240 Computing for Data Analysis - Homework 3

Name:
GTID:

Deadline: 11:59 pm EDT, March 30

- Submit your answers as one single PDF file on Gradescope.
- You will be allowed 2 total late days (48 hours) without penalty for the entire semester. Once those days are used, you will be penalized according to the following policy:
  - Homework is worth full credit before the due time.
  - It is worth 75% credit for the next 24 hours.
  - It is worth 50% credit for the second 24 hours.
  - It is worth zero credit after that.
- You are required to use Latex, or word processing software, to generate your solutions to the written questions. Handwritten solutions WILL NOT BE ACCEPTED.

1 Naive Bayes: Linear Model in Disguise [45 pts]

In this problem set, we will demonstrate that a Bernoulli Naive Bayes classifier is essentially a linear classifier. Specifically, we will show that it shares the same functional form as Logistic Regression.

Consider a binary classification problem with classes $C \in \{0, 1\}$ and a vector of n binary features $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ where $x_i \in \{0, 1\}$. A classifier intends to predict the class C of a given data point $\boldsymbol{x}$, according to the decision rule:

$$\text{The class of } \boldsymbol{x} \text{ is } \begin{cases} C = 1 & \text{if } \dfrac{P(C = 1|\boldsymbol{x})}{P(C = 0|\boldsymbol{x})} > 1 \\ C = 0 & \text{otherwise.} \end{cases} \tag{1}$$

We assume the class-conditional probabilities for each feature i is given:

$$\begin{aligned} P(x_i = 0|C = 0) &= \alpha_i \\ P(x_i = 0|C = 1) &= \beta_i \end{aligned} \tag{2}$$

where $\alpha_i, \beta_i \in [0, 1]$. Besides, the class prior probabilities are also given:

$$\begin{aligned} P(C = 0) &= c \\ P(C = 1) &= 1 - c \end{aligned} \tag{3}$$

Answer the following questions:

1. **[15 points]** Demonstrate that the term $\ln \frac{P(x_i|C=1)}{P(x_i|C=0)}$ can be written as a linear function of the component x_i. Write out the expression of the weight w_i and the bias b_i in terms of α_i, β_i.

2. **[10 points]** Apply Bayes' theorem and Naive Bayes assumption, demonstrate that $\ln \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})}$ can be expressed as $\boldsymbol{w}^\top \boldsymbol{x} + w_0$. Write out the expression of $\boldsymbol{w}$ and w_0 in terms of $\{\alpha_i\}_{i=1}^n, \{\beta_i\}_{i=1}^n$ and c. Explicitly mark where you applied the Naive Bayes assumption.

3. **[10 points]** Recall that the Logistic Regression predicts the probability of a class using the sigmoid function:

$$P_{\text{LR}}(C=1|\boldsymbol{x}) = \frac{1}{1+\exp\left(-(\boldsymbol{w}^\top \boldsymbol{x} + w_0)\right)} \tag{4}$$

   Using your result from the last question, show that the Naive Bayes posterior probability $P(C=1|\boldsymbol{x})$ can be written in this exact same functional form.

4. **[10 pts]** Naive Bayes (Generative) and Logistic Regression (Discriminative) both result in linear decision boundaries but differ significantly in execution. Please compare these models (e.g., in terms of their learning algorithms, modeling assumptions, and etc.), and briefly discuss the benefits and weaknesses of Naive Bayes.

Solution:

1. For a binary feature $x_i \in \{0,1\}$, the probability $P(x_i|C)$ can be written as:

$$P(x_i|C=0) = \alpha_i^{(1-x_i)}(1-\alpha_i)^{x_i}, \quad P(x_i|C=1) = \beta_i^{(1-x_i)}(1-\beta_i)^{x_i} \tag{5}$$

   The log-ratio is:

$$\ln \frac{P(x_i|C=1)}{P(x_i|C=0)} = \ln \frac{\beta_i^{(1-x_i)}(1-\beta_i)^{x_i}}{\alpha_i^{(1-x_i)}(1-\alpha_i)^{x_i}} = (1-x_i)\ln \frac{\beta_i}{\alpha_i} + x_i \ln \frac{1-\beta_i}{1-\alpha_i} \tag{6}$$

   Rearranging to group x_i terms:

$$\ln \frac{P(x_i|C=1)}{P(x_i|C=0)} = x_i \left(\ln \frac{1-\beta_i}{1-\alpha_i} - \ln \frac{\beta_i}{\alpha_i} \right) + \ln \frac{\beta_i}{\alpha_i} \tag{7}$$

   Thus, $w_i = \ln \frac{\alpha_i(1-\beta_i)}{\beta_i(1-\alpha_i)}$ and $b_i = \ln \frac{\beta_i}{\alpha_i}$.

2. Using Bayes' Theorem:

$$\frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})} = \frac{P(\boldsymbol{x}|C=1)P(C=1)}{P(\boldsymbol{x}|C=0)P(C=0)} \tag{8}$$

   Applying the **Naive Bayes Assumption** (features are independent given the class):

$$P(\boldsymbol{x}|C) = \prod_{i=1}^{n} P(x_i|C) \tag{9}$$

Taking the natural log:

$$\ln \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})} = \ln \frac{P(C=1)}{P(C=0)} + \sum_{i=1}^{n} \ln \frac{P(x_i|C=1)}{P(x_i|C=0)} \tag{10}$$

Substituting the result from Part 1:

$$\ln \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})} = \ln \frac{1-c}{c} + \sum_{i=1}^{n} (w_i x_i + b_i) = \sum_{i=1}^{n} w_i x_i + \left(\ln \frac{1-c}{c} + \sum_{i=1}^{n} b_i \right) \tag{11}$$

Where $w_i = \ln \frac{\alpha_i(1-\beta_i)}{\beta_i(1-\alpha_i)}$ and $w_0 = \ln \frac{1-c}{c} + \sum_{i=1}^{n} \ln \frac{\beta_i}{\alpha_i}$.

3. Let $a = \ln \frac{P(C=1|\boldsymbol{x})}{P(C=0|\boldsymbol{x})} = \boldsymbol{w}^\top \boldsymbol{x} + w_0$. Then $\frac{P(C=1|\boldsymbol{x})}{1-P(C=1|\boldsymbol{x})} = \exp(a)$. Solving for $P(C=1|\boldsymbol{x})$:

$$P(C=1|\boldsymbol{x}) = \frac{\exp(a)}{1+\exp(a)} = \frac{1}{1+\exp(-a)} = \frac{1}{1+\exp(-(\boldsymbol{w}^\top \boldsymbol{x} + w_0))} \tag{12}$$

This matches the Sigmoid function used in Logistic Regression.

4.
- **Learning:** Naive Bayes (NB) estimates parameters via counting; Logistic Regression (LR) optimizes weights via Gradient Descent.
- **Assumptions:** NB assumes conditional independence of features. LR assumes a general linear relationship between features and the log-odds.
- **Pros/Cons of NB:** NB converges faster and works well with small datasets, but it is often a "bad estimator" of actual probabilities and suffers if the independence assumption is strongly violated.

2 K-Means Clustering [55 pts]

In this exercise, you will implement the K-means clustering algorithm from scratch to partition data into clusters. We have prepared a Jupyter notebook to guide you through the complete K-means workflow.

Please access the notebook using this link. You can also download the notebook by using `File` (top-left corner) → `Download` → `Download .ipynb`.

For this problem, complete all `5` sections in the notebook, execute all cells to generate the required outputs, and export the notebook into a single PDF file. For example, if you are using Google Colab, you can use `File` (top-left corner) → `Print` to generate a PDF. For submission, a separate homework entry for this problem will be created on GradeScope. Please upload the generated PDF to that entry.

Solution:
Solution Notebook Link

References

Solution:
Please mention any AI tools, people, post or blog etc. you used.