Machine Learning Basics

Instructors: Mohammadreza A. Dehaqani, Babak n. Arabi, Mostafa Tavassolipour

Aidin Kazemi, Taha Majlesi

Fall 2025

HomeWork 1

Question 1: Bayes Rule vs. Random Decisions

Suppose the task is to classify the input signal x into one of K classes $\omega \in \{1, 2, \ldots, K\}$ such that the action $\alpha(x) = i$ means classifying x into class i. The Bayesian decision rule is to maximize the posterior probability

$$\alpha_{Bayes}(x) = \omega^* = \arg\max_i p(\omega_i|x).$$

Suppose we replace it by a randomized decision rule, which classifies x to class i following the posterior probability $p(\omega_i|x)$, i.e.,

$$\alpha_{rand}(x) = \omega \sim p(\omega|x).$$

1. What is the overall risk R_{rand} for this decision rule? Derive it in terms of the posterior probability using the zero-one loss function.
2. Show that this risk R_{rand} is always no smaller than the Bayes risk R_{Bayes}. Thus, we cannot benefit from the randomized decision.
3. Under what conditions on the posterior are the two decision rules the same?

Question 2: Bayes Boundaries for Gaussian Classes

Suppose we have a two-class recognition problem with salmon ($\omega = 1$) and sea bass ($\omega = 2$).

1. First, assume we have one feature, the pdfs are the Gaussians $N(0, \sigma^2)$ and $N(1, \sigma^2)$ for the two classes, respectively. Show that the threshold τ minimizing the average risk is equal to

$$\tau = \frac{1}{2} - \sigma^2 \ln \frac{\lambda_{12} P(\omega_2)}{\lambda_{21} P(\omega_1)}$$

   where we have assumed $\lambda_{11} = \lambda_{22} = 0$.

2. Next, suppose we have two features $\mathbf{x} = (x_1, x_2)$ and the two class-conditional densities, $p(\mathbf{x}|\omega = 1)$ and $p(\mathbf{x}|\omega = 2)$, are 2D Gaussian distributions centered at points $(4, 11)$ and $(10, 3)$ respectively with the same covariance matrix $\Sigma = 3I$ (where I is the identity matrix). Suppose the priors are $P(\omega = 1) = 0.6$ and $P(\omega = 2) = 0.4$.
   (a) Suppose we use a Bayes decision rule, write the two discriminant functions $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$.
   (b) Derive the equation for the decision boundary $g_1(\mathbf{x}) = g_2(\mathbf{x})$.
   (c) How would the decision boundary change if we changed the priors? the covariances?
   (d) Using computer software, sample 100 points from each of the two densities. Draw the boundary on the feature space (the 2D plane).

Question 3: Naive Bayes Parameter Estimation (MLE)

The goal is to find the Maximum Likelihood Estimates (MLEs) for the parameters $\phi = \{\phi_y, \phi_{j|y=0}, \phi_{j|y=1}\}$ by maximizing the joint log-likelihood $\ell(\phi)$. Model Parameters for Bernoulli Distribution are:

- Prior Probability:

$$\phi_y = P(y = 1)$$

- Feature Likelihoods:

$$\phi_{j|y=0} = P(x_j = 1|y = 0)$$
$$\phi_{j|y=1} = P(x_j = 1|y = 1)$$

1. Write the Joint Log-Likelihood Function $\ell(\phi)$.
2. Show that the MLEs correspond to the empirical frequencies.

Question 4: Modeling Goals with MLE

You are the Reign FC manager, and the team is five games into its 2021 season. The number of goals scored by the team in each game so far are given below:

$$[2, 4, 6, 0, 1].$$

Let's call these scores $x_1, \ldots, x_5$. Based on your (assumed iid) data, you'd like to build a model to understand how many goals the Reign are likely to score in their next game. You decide to model the number of goals scored per game using a Poisson distribution. Recall that the Poisson distribution with parameter λ assigns every non-negative integer $x = 0, 1, 2, \ldots$ a probability given by

$$\text{Poi}(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$$

1. Derive an expression for the maximum-likelihood estimate of the parameter λ governing the Poisson distribution in terms of goal counts for the first n games: $x_1, \ldots, x_n$. (Hint: remember that the log of the likelihood has the same maximizer as the likelihood function itself.)
2. Give a numerical estimate of λ after the first five games. Given this λ, what is the probability that the Reign score 6 goals in their next game?
3. Suppose the Reign score 8 goals in their 6th game. Give an updated numerical estimate of λ after six games and compute the probability that the Reign score 6 goals in their 7th game.

Question 5: MLE Derivation of the Least Squares Loss

In **Linear Regression**, we assume the relationship between input $\mathbf{x}^{(i)}$ and output $y^{(i)}$ is $y^{(i)} = \boldsymbol{\theta}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$, where the error $\epsilon^{(i)}$ is independent and follows a **Gaussian (Normal) distribution** with mean zero and variance σ^2: $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$.

1. Write the likelihood $p(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}, \sigma^2)$.
2. Show that maximizing the log-likelihood $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is equivalent to minimizing the squared error cost function (Mean Squared Error).

Question 6: Effect of Priors on the Decision Boundary

Consider a two-class, one-dimensional problem ($\mathbf{x} = x, \omega_1, \omega_2$) where the likelihoods are standard Gaussians with equal variance σ^2 but different means:

$$p(x|\omega_1) \sim \mathcal{N}(\mu_1, \sigma^2) \quad \text{and} \quad p(x|\omega_2) \sim \mathcal{N}(\mu_2, \sigma^2)$$

Derive the decision boundary condition (x^*) if the **priors are unequal**: $P(\omega_1)$ and $P(\omega_2)$. Explain how the value of x^* shifts as $P(\omega_1)$ increases relative to $P(\omega_2)$.

Question 7: Naive Bayes Warm-Up

In this problem, we'll first review a standard way that Bayes rule is used and then explore implementing a Naive Bayes classifier, a very fast classification method that is often surprisingly accurate for text data with simple representations like bag of words.

1. Suppose a drug test produces a positive result with probability 0.99 for drug users, $P(T=1|D=1)=0.99$. It also produces a negative result with probability 0.99 for non-drug users, $P(T=0|D=0)=0.99$. The probability that a random person uses the drug is 0.001, so $P(D=1)=0.001$. What is the probability that a random person who tests positive is a user, $P(D=1|T=1)$?

2. Consider the dataset below, which has 10 training examples and 2 features:

$$\mathbf{X}=\begin{bmatrix}0&1\\1&1\\0&0\\1&1\\1&1\\0&0\\1&0\\1&0\\1&1\\1&0\end{bmatrix},\quad \mathbf{y}=\begin{bmatrix}1\\1\\0\\1\\1\\0\\0\\0\\1\\0\end{bmatrix}.$$

   Suppose you believe that a naive Bayes model would be appropriate for this dataset, and you want to classify the following test example:

$$\mathbf{x}=\begin{bmatrix}1&1\end{bmatrix}.$$

   (a) Compute the estimates of the class prior probabilities:

   - $P(y=1)$.
   - $P(y=0)$.

   (b) Compute the estimates of the 4 conditional probabilities required by Naive Bayes for this example:

   - $P(x_1=1|y=1)$.
   - $P(x_2=1|y=1)$.
   - $P(x_1=1|y=0)$.
   - $P(x_2=1|y=0)$.

   (c) Under the naive Bayes model and your estimates of the above probabilities, what is the most likely label for the test example? (Show your work.)

Provide your implementation in a well-structured ***Jupyter Notebook*** *(`.ipynb`) with clear markdown explanations and organized code cells.*

Question 8: Classifying Spam with Naive Bayes

1. The dataset we will be using is a subset of 2005 TREC Public Spam Corpus, containing 9000 training examples and 1000 test examples. You can download it here. Each line in the train/test files represents a single email with the following space-delimited properties: the first is the email ID (in the form /xxx/yyy), the second is whether it is 'spam' or 'ham' (non-spam), and the rest are words followed by their occurrence numbers. (Note that numbers may be words, so don't worry if a line contains multiple numbers in a row). The data has been pre-processed to remove non-word characters (e.g. '¡) and to select features similar to what Mehran Sahami did in his original paper, though with larger cut-offs since our corpus is larger.

2. Using the training data, compute the prior probabilities $P(\text{spam})$ and $P(\text{ham})$. What is $P(\text{spam})$?

3. Determine the vocabulary and compute the conditional probabilities $P(w_i|\text{spam})$ and $P(w_i|\text{ham})$ using the m-estimate discussed in class (with $m=|\text{Vocabulary}|$ and $p=1/|\text{Vocabulary}|$). In this context we consider each word as a training example, so n is the total number of words (in either ham or spam

documents) and n_i is the number of times w_i appeared in those documents (including multiple occurrences in the same email). What are the 5 most likely words given that a document is spam? What are the 5 most likely words given that a document is ham?

4. Use these probabilities to classify the test data and report the accuracy (i.e. the percentage of correct classifications). Note that directly computing $P(\text{spam}|w_1,\ldots,w_n)$ and $P(\text{ham}|w_1,\ldots,w_n)$ can cause numerical precision issues, since the unnormalized probabilities are very small (i.e. the numerator in Bayes' theorem). Instead, you should compare the log-probabilities of being ham/spam.

5. Vary the m parameter, using $m = |\text{Vocabulary}| \times [1, 10, 100, 1000, 10000]$ and plot the accuracies vs. m. What assumptions are we making when the value of m is very large vs. very small? How does this affect the test accuracy?

6. If you were a spammer, how would you modify your emails to beat the classifiers we have learned above?

Provide your implementation in a well-structured ***Jupyter Notebook*** *(`.ipynb`) with clear markdown explanations and organized code cells.*

Question 9: Cloud vs. Clear Sky

A dataset of cloud and clear sky images has been collected. Each image was either tagged c (cloudy sky) or s (clear sky).

1. Define a criterion for separating the images based on their features, and implement your classification algorithm (Do not use well-known classification algorithms and instead, design your own rule based on simple features such as color).

2. Test the algorithm on the dataset and report the Confusion Matrix, Precision, and Recall.

3. Explore and define new classifiers (again, based on new criterions you define) to improve the performance.

4. Show and discuss the misclassified images and causes of misclassification according to your defined criteria.

5. If you used multiple models, compare the accuracy and misclassified images for each case.

Provide your implementation in a well-structured ***Jupyter Notebook*** *(`.ipynb`) with clear markdown explanations and organized code cells.*