Machine Learning Assignment 7

(Bayesian Classifiers & Ensemble)

Due: 9, June

1. [20pts] Naïve Bayes Classifier

Suppose you are given the following set of data with four Integer input variables A, B, C, and D, and a single binary label y.

In this task, the value of a variable means how many times it appears in text from the corresponding label (i.e., word frequency, which is a popular representation of text). For example, in the text of x_1 from label +1, word A appears twice, word B appears 4 times, word C appears 10 times and word D appears 3 times.

We are trying to fit a Naïve Bayes Classifier on this dataset.

	A	B	C	D	y
x_1	2	4	10	3	+1
x_2	3	1	4	2	+1
x_3	0	2	0	5	-1
x_4	2	0	4	0	+1
x_5	1	6	6	0	-1
x_6	0	2	1	7	-1
x_7	3	0	0	8	+1
x_8	6	1	2	7	-1

(1) [**10pts**] Calculate the empirical conditional probability of each variable for appearing in texts from each label. To illustrate, for variable A, calculate $p_{A,j} = P(word = A \mid y = j),\ j \in \{-1, +1\}$, and the same for the other variables. Remember to use Laplace smoothing to avoid zero probabilities.

(2) [**10pts**] Give a new sample where $A = 3, B = 2, C = 1, D = 2$. Predict its label. You should write down your calculation in detail. (It is enough to only give the form of a fraction, not necessarily calculated as a decimal, 仅给出分数形式即可，不一定需要计算为小数)

(1) From Laplacian correction formula,

$$\hat{P}(x_i \mid c) = \frac{\mid D_{c,x_i} \mid +1}{\mid D_c \mid +N_i}$$

Where $|D_c|$ is the number of the set of c^{th} class samples in the training set D, and for discrete features, let $|D_{c,x_i}|$ denotes the number of the set of samples that take the value x_i on the i^{th} feature in D_c, besides, N_i represents the number of possible values of the i^{th} feature.

In our example, $|D_{+1}| = |D_{-1}| = 46,\ N_i = 4.$

We can obtain the empirical conditional probability of each variable

$$P_{A,+1} = \frac{10+1}{46+4} = \frac{11}{50}, P_{A,-1} = \frac{7+1}{46+4} = \frac{8}{50}$$

$$P_{B,+1} = \frac{5+1}{46+4} = \frac{6}{50}, P_{B,-1} = \frac{11+1}{46+4} = \frac{12}{50}$$

$$P_{C,+1} = \frac{18+1}{46+4} = \frac{19}{50}, P_{C,-1} = \frac{9+1}{46+4} = \frac{10}{50}$$

$$P_{D,+1} = \frac{13+1}{46+4} = \frac{14}{50}, P_{D,-1} = \frac{19+1}{46+4} = \frac{20}{50}$$

(2) Let's calculate the class prior probability first. From Laplacian correction formula,

$$\hat{P}(c) = \frac{\mid D_c \mid +1}{\mid D \mid +N}$$

Where $|D|$ is the number of the dataset, and N represents the number of possible categories. In our example, $|D| = 92$ and $N = 2$, then we have

$$\hat{P}(y=+1) = \hat{P}(y=-1) = \frac{46+1}{92+2}$$

Therefore, based on (1), we have

$$P(y=+1|A=3, B=2, C=1, D=2) \propto \left(\frac{11}{50}\right)^3 \times \left(\frac{6}{50}\right)^2 \times \frac{19}{50} \times \left(\frac{14}{50}\right)^2 = \frac{1331 \times 36 \times 166}{50^8}$$

$$P(y=-1|A=3, B=2, C=1, D=2) \propto \left(\frac{8}{50}\right)^3 \times \left(\frac{12}{50}\right)^2 \times \frac{10}{50} \times \left(\frac{20}{50}\right)^2 = \frac{1536 \times 40 \times 400}{50^8}$$

Due to $P(y=-1|A=3, B=2, C=1, D=2) > P(y=+1|A=3, B=2, C=1, D=2)$, its label is -1.

2. [35pts] Gaussian Bayesian Classifiers

Given data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, where $y \in Y = \{1,2,\dots,K\}$.

(1) [**5pts**] Please write down the Bayes optimal classifier that minimizes the misclassification error rate.

(2) [**15pts**] Suppose the samples in the k-th class are i.i.d. sampled form normal distribution $\mathcal{N}(\mu_k, \Sigma)$, ($k = 1, 2, \dots$, K, all classes share the same covariance matrix). Let m_k denote the number of samples in the k-th class, and the prior probability $P(y = k) = \pi_k$. If $x \in R^d \sim \mathcal{N}(\mu, \Sigma)$, then the probability density function is:

$$p(x) = \frac{1}{(2\pi)^{d/2}\, det(\Sigma)^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right)$$

Please write down the corresponding Bayes optimal classifier.

(3) [**15pts**] For binary classification problem, please prove that when samples in each class are i.i.d. sampled from normal distributions which share the same covariance matrix and the two classes have equal prior probabilities $\pi_0 = \pi_1$, LDA (Linear Discriminant Analysis) gives the Bayes optimal classifier.

Hint: The optimal solution of LDA is:

$$w = S_w^{-1}(\mu_0 - \mu_1)$$

where S_w is within-class scatter matrix, $S_w = \Sigma_0 + \Sigma_1$ (Σ_i is the covariance matrix of the i-th class).

(1) If we use 0-1 loss, the conditional risk is $R(c|\boldsymbol{x}) = 1 - P(c|\boldsymbol{x})$. The Bayes optimal classifier that minimizes the misclassification error rate is

$$h^*(x) = \arg\min_{x \in \mathcal{Y}} R(c|\boldsymbol{x}) \Leftrightarrow \arg\max_{x \in \mathcal{Y}} P(c|\boldsymbol{x})$$

(2)

$$\begin{aligned}
h^*(\boldsymbol{x}) &= \arg\max_y P(\,y \mid \boldsymbol{x}\,) \\
&= \arg\max_y \ln P(\,y \mid \boldsymbol{x}\,) \\
&= \arg\max_y \ln P(y)\, P(\,\boldsymbol{x} \mid y\,) \\
&= \arg\max_y \ln \pi_y + \ln P(\,\boldsymbol{x} \mid y\,) \\
&= \arg\max_y \ln \pi_y - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_y)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_y)
\end{aligned}$$

$$= \arg\max_{y} \ln \pi_y - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_y - \frac{1}{2}\boldsymbol{\mu}_y^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_y$$

Note that we can omit $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ because it's not related to y.

(3) From the meaning of the question, we want to prove that Bayesian most classifier and LDA have equivalent effects. Recall what I learned in pattern recognition, the discriminant function of a binary classification problem can be written as

$$\begin{aligned} &g(x) = P(w_1|x) - P(w_2|x) \\ &= \ln\frac{P(x|w_1)}{P(x|w_2)} + \ln\frac{P(w_1)}{P(w_2)} \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mu_0 - \mu_1) - \frac{1}{2}\left(\boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right) + \ln\left(\frac{\pi_0}{\pi_1}\right) \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mu_0 - \mu_1) - \frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) + \ln\left(\frac{\pi_0}{\pi_1}\right) \end{aligned}$$

When the two classes share the same covariance matrix, The optimal solution of LDA is:

$$\begin{aligned} w &= S_w^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \\ &= (\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \\ &= (2\boldsymbol{\Sigma})^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \end{aligned}$$

where S_w is within-class scatter matrix, $S_w = \Sigma_0 + \Sigma_1$ (Σ_i is the covariance matrix of the i-th class).

The midpoint of the projected class centers is

$$\boldsymbol{c} = \frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w} = \frac{1}{4}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1).$$

The decision boundary of LDA is

$$\begin{aligned} f(x) &= x^{\mathrm{T}}w - c \\ &= \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) - \frac{1}{4}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \end{aligned}$$

Since $\pi_0 = \pi_1$, therefore $g(x) = \frac{1}{2}f(x)$, but they are equivalent.

3. [30pts] MLE and Linear Regression

Sample points come from an unknown distribution, $X_i \sim D$. Labels y_i are the sum of a deterministic function $f(\mathrm{X_i})$ plus random noise: $y_i = f(X_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

For this problem, we will assume that $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$—that is, the variance σ_i^2 of the noise is different for each sample point and we will examine how our loss function changes as a result. We assume that we know the value of each σ_i^2. You are given an $n \times p$ design matrix X, an n-dimensional vector y of labels, such that the label y_i of sample point X_i is generated as described above, and a list of the noise variances σ_i^2.

(1) [**10pts**] Apply MLE to derive the optimization problem that will use the maximum likelihood estimate of the distribution parameter f. (Note: f is a function, but we can still treat it as the parameter of an optimization problem.) Express your Objective function as a summation of loss functions, one per sample point.

(2) [**10pts**] We decide to do linear regression, so we parameterize $f(X_i)$ as $f(X_i) = w \cdot X_i$, where w is a p-dimensional vector of weights. Write an equivalent optimization problem where your optimization variable is w and the cost function is a function of X, y, w, and the variances σ_i^2. Find a way to express your cost function in matrix notation. (Hint: You can define a new matrix.)

(3) [**10pts**] Write the solution to your optimization problem as the solution of a linear system of equations. (Again, in matrix notation.)

(1) By applying MLE, we can obtain maximum likelihood function

$$\ell = \sum_i \ln \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{1}{2}\left(\frac{y_i - f(X_i)}{\sigma_i}\right)^2\right)$$

$$= \sum_i \ln \frac{1}{\sqrt{2\pi\sigma_i}} - \frac{1}{2}\sum_i \left(\frac{y_i - f(X_i)}{\sigma_i}\right)^2$$

Therefore, after ignoring irrelevant constants, the loss function is

$$\mathcal{L} = \frac{1}{2}\sum_i \left(\frac{y_i - f(X_i)}{\sigma_i}\right)^2$$

(2) Easy,

$$\mathcal{L} = \frac{1}{2}\sum_i \left(\frac{y_i - f(X_i)}{\sigma_i}\right)^2$$

$$= \frac{1}{2}\sum_i \frac{1}{\sigma_i^2}(y_i - w \cdot X_i)^2$$

$$= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top \boldsymbol{\Sigma}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

Where $\boldsymbol{\Sigma} = diag(\sigma_1^{-2}, \sigma_2^{-2}, \cdots, \sigma_n^{-2})$

(3) Same as linear regression, we can obtain its close form solution by letting

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{X}^\top \boldsymbol{\Sigma}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = 0$$

Then,

$$\boldsymbol{w} = (\boldsymbol{X}^{\top}\boldsymbol{\Sigma}\boldsymbol{X})^{-\mathbf{1}}\boldsymbol{X}^{\top}\boldsymbol{\Sigma}\boldsymbol{y}$$

4. [5pts] Bagging & Random Forest, Short answers

Bagging with decision trees as base learners, and Random Forest, which one has the lower computational cost? And state the reason. (Assume they are trained the same number of iterations)

Ans:

Random Forest has a higher computational cost compared to bagging with decision trees because it includes an additional step of feature randomization during training, which requires evaluating optimal split points for randomly selected features.