ECE M146 Homework 5 Solution
Introduction to Machine Learning Monday, May 3, 2021
Instructor: Lara Dolecek Due: Monday, May 10, 2021
TA: Zehui (Alex) Chen chen1046@ucla.edu

Please upload your homework to Gradescope by May 10, 4:00 pm.
Please submit a single PDF directly on Gradescope
You may type your homework or scan your handwritten version. Make sure all the work is discernible.

1. In this problem, we are going to implement the kernelized perceptron algorithm to see the usefulness of the kernel trick. Suppose we have a training dataset contains N samples with feature $x_i \in \mathbb{R}^{n+1}$ and label $y_i \in \{-1, 1\}$. Note that the feature is augmented with an additional feature 1 to take care of the bias term.

   To implement the kernalized perceptron algorithm, we only need to make the following modifications to the original perceptron algorithm:

   (a) Instead of finding the weight w, we are trying to find dual variables $a_1, a_2, \cdots, a_N$. Over here, a_i is the error counter for sample i.

   (b) Looping through the data as usual, when a sample i is misclassified, increment the error counter, i.e., $a_i := a_i + 1$.

   (c) Whenever you need to make a prediction on a point x_{test}, e.g., the 5-th line of Algorithm 5 in the textbook, replace the prediction with:

   $$\sum_{i=1}^{N} a_i y_i K(x_{test}, x_i),$$

   where $K(u, v)$ is a kernel function. This calculation is the same as $w^T x_{test}$ if no kernel is used.

   We are going to use the following three kernels in this problem:

   - "No kernel" kernel: $K(u, v) = u^T v$.
   - Polynomial kernel with degree 2: $K(u, v) = (1 + u^T v)^2$.
   - Gaussian kernel: $K(u, v) = \exp(-\sigma \|u - v\|^2)$, where σ is a user defined parameter.

   With the kernalized perceptron algorithm, we are going to get non-linear decision boundaries. Because of this, we need to use a contour plot for visualizing the decision boundary. Pseudo-code for visualization is as follows based on our running data set.

Create mesh grids GPA and GRE with `meshgrid` (`np.meshgrid` in python) within the range of $0 - 4$.
Calculate a matrix Z whose element Z_{ij} is the perdition of your model using GPA_{ij} and GRE_{ij}, e.g., $w^T[GPA_{ij}, GRE_{ij}, 1]^T$ if no kernel is used.
Generate the contour plot using GPA, GRE and Z and set `LevelList` (`level` in python) to 0.

Perform the following tasks:

(a) Implement the kernelized perceptron algorithm using the "No kernel" kernel. Train the model on the *UCLA_EE_grad_2031.csv* dataset and loop through the dataset for 1000 iterations. With the "No kernel" kernel, you can find $w = \sum_{i=1}^{N} a_i x_i y_i$. Plot the decision boundary, report the training accuracy, and report the w. This is a sanity check for your implementation. If you implement it correctly, you should get the same results as in HW1.

(b) Implement the kernelized perceptron algorithm using the polynomial kernel of degree 2. Train the model on the *UCLA_EE_grad_2031.csv* dataset and loop through the dataset for 1000 iterations. Plot the decision boundary and report the training accuracy.

(c) Implement the kernelized perceptron algorithm using the Gaussian kernel with $\sigma = 1$. Train the model on the *UCLA_EE_grad_2031.csv* dataset and loop through the dataset for 1000 iterations. Plot the decision boundary and report the training accuracy.

(d) Training on linearly separable dataset does not show the powerfulness of the kernel trick. Now training on the *UCLA_EE_grad_2030.csv* dataset and loop through the dataset for 1000 iterations. Plot the decision boundary and report the training accuracy using the polynomial kernel of degree 2, Gaussian kernel with $\sigma = 1$, and Gaussian kernel with $\sigma = 3$, respectively. What do you observe? With which kernel can you get 100% training accuracy?

2. Consider 3 random variables A,B and C with joint probabilities $P(A, B, C)$ listed in the following table.

	C=0		C=1	
	B=0	B=1	B=0	B=1
A=0	0.096	0.024	0.27	0.03
A=1	0.224	0.056	0.27	0.03

(a) Calculate $P(A|C = 0)$, $P(B|C = 0)$, and $P(A, B|C = 0)$.

(b) Calculate $P(A|C = 1)$, $P(B|C = 1)$, and $P(A, B|C = 1)$.

(c) Is A conditionally independent of B given C?

(d) Calculate $P(A)$, $P(B)$, and $P(A, B)$.

(e) Is A independent of B?

3. Let us revisit the restaurant selection problem in HW3. You are trying to choose between two restaurants (sample 9 and sample 10) to eat at. To do this, you will train a classifier based on your past experiences (sample 1-8). The features for each restaurants and your judgment on the goodness of sample 1-8 are summarized by the following chart.

Sample #	HasOutdoorSeating	HasBar	IsClean	HasGoodAtmosphere	IsGoodRestaurant
1	0	0	0	1	1
2	1	1	0	0	0
3	0	1	1	1	1
4	1	0	0	1	1
5	1	1	1	0	0
6	1	0	1	0	1
7	1	1	0	1	1
8	0	0	1	1	1
9	0	1	0	1	?
10	1	1	1	1	?

In this exercise, instead of a decision tree, you will use the Naïve Bayes classifier to decide whether restaurant 9 and 10 are good or not. For clarity, we abbreviate the names of the features and label as follows: HasOutdoorSeating $\rightarrow O$, HasBar $\rightarrow B$, IsClean $\rightarrow C$, HasGoodAtmosphere $\rightarrow A$, and IsGoodRestaurant $\rightarrow G$.

(a) Train the Naïve Bayes classifier by calculating the maximum likelihood estimate of class priors and class conditional distributions. Namely, calculate the maximum likelihood estimate of the following: $P(G)$, and $P(X|G), X \in \{O, B, C, A\}$.

(b) For Sample #9 and #10, make the decision using

$$\hat{G}_i = \underset{G_i \in \{0,1\}}{\operatorname{argmax}} \quad P(G_i)P(O_i, B_i, C_i, A_i|G_i),$$

where O_i, B_i, C_i, and A_i are the feature values for the i-th sample.

(c) We use Laplace smoothing to avoid having class conditional probabilities that are strictly 0. To use Laplace smoothing for a binary classifier, add 1 to the numerator and add 2 to the denominator when calculating the class conditional distributions. Let us re-calculate the class conditional distributions with Laplace smoothing. Namely, calculate the maximum likelihood estimate of $P(X|G), X \in \{O, B, C, A\}$.

(d) Repeat (b) with the class conditional distributions you get from (c).

4. In class, we learned a Naïve Bayes classifier for binary feature values, i.e., $x_j \in \{0, 1\}$ where we model the class conditional distribution to be Bernoulli. In this exercise, you are going to extend the result to the case where features that are non-binary.

   We are given a training set $\{(x^{(i)}, y^{(i)}); i = \{1, \cdots, m\}\}$, where $x^{(i)} \in \{1, 2, \cdots, s\}^n$ and $y^{(i)} \in \{0, 1\}$. Again, we model the label as a biased coin with $\theta_0 = P(y^{(i)} = 0)$ and $1 - \theta_0 = P(y^{(i)} = 1)$. We model each non-binary feature value $x_j^{(i)}$ (an element of $x^{(i)}$) as a biased dice for each class. This is parameterized by:

$$
\begin{aligned}
&P(x_j = k|y = 0) = \theta_{j,k|y=0},\ k = 1, \cdots, s-1;\\
&P(x_j = s|y = 0) = \theta_{j,s|y=0} = 1 - \sum_{k=1}^{s-1} \theta_{j,k|y=0};\\
&P(x_j = k|y = 1) = \theta_{j,k|y=1},\ k = 1, \cdots, s-1;\\
&P(x_j = s|y = 1) = \theta_{j,s|y=1} = 1 - \sum_{k=1}^{s-1} \theta_{j,k|y=1};
\end{aligned}
$$

   Notice that we do not model $P(x_j = s|y = 0)$ and $P(x_j = s|y = 1)$ directly. Instead we use the above equations to guarantee all probabilities for each class sum to 1.

   (a) Using the **Naïve Bayes (NB) assumption**, write down the joint probability of the data:

$$P(x^{(1)}, \cdots, x^{(m)}, y^{(1)}, \cdots, y^{(m)})$$

   in terms of the parameters θ_0, $\theta_{j,k|y=0}$ and $\theta_{j,k|y=1}$. You may find the indicator function $\mathbf{1}(\cdot)$ useful.

   (b) Now, maximize the joint probability you get in (a) with respect to each of θ_0, $\theta_{j,k|y=0}$, and $\theta_{j,k|y=1}$. Write down your resulting θ_0, $\theta_{j,k|y=0}$ and $\theta_{j,k|y=1}$ and show intermediate steps. Explain in words the meaning of your results.

5. The pdf for two jointly Gaussian random variables X and Y is of the following form parameterized by the scalars m_1, m_2, σ_1, σ_2 and ρ_{XY}:

$$f_{X,Y}(x,y) = \frac{\exp\left\{\frac{-1}{2(1-\rho_{XY}^2)}\left[(\frac{x-m_1}{\sigma_1})^2 - 2\rho_{XY}\left(\frac{x-m_1}{\sigma_1}\right)\left(\frac{y-m_2}{\sigma_2}\right) + (\frac{y-m_2}{\sigma_2})^2\right]\right\}}{2\pi\sigma_1\sigma_2\sqrt{1-\rho_{XY}^2}}. \quad (1)$$

The pdf for multivariate jointly Gaussian random variable $Z \in \mathbb{R}^k$ is of the following form parameterized by $\mu \in \mathbb{R}^k$ and $\Sigma \in \mathbb{R}^{k\times k}$.

$$f_Z(z) = \frac{\exp\left\{-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)\right\}}{\sqrt{(2\pi)^k|\Sigma|}}. \quad (2)$$

Suppose $Z = [X,Y]^T$, i.e., $z = [x,y]^T$.

(a) Find μ, Σ^{-1} and Σ in terms of m_1, m_2, σ_1, σ_2 and ρ_{XY}.

(b) Suppose $\rho_{XY} = 0$, what is Σ in this case? Can you write $f_{X,Y}(x,y)$ as the product of two single variate Gaussian distributions? Are X and Y independent?

6. The Gaussian Discriminant Analysis (GDA) models the class conditional distribution as multivariate Gaussian, i.e, $P(X|Y) \sim \mathcal{N}(\mu_Y, \Sigma)$. Suppose we want to enforce the **Naive Bayes (NB) assumption**, i.e. $P(X_i|Y, X_j) = P(X_i|Y), \forall j \neq i$, where X_i and X_j are elements of random vector X, to GDA. Show that all off diagonal elements of Σ equal to 0: $\Sigma_{i,j} = 0, \forall i \neq j$ with the **NB assumption**.

7. In this exercise, we will implement a binary classifier using the Gaussian Discriminant Analysis (GDA) model in MATLAB (or python) with the *UCLA_EE_grad_2030.csv* data. The first two columns are the feature values and the last column contains the class labels. In this homework, we will implement the GDA for the scalar case. Because of this, we will need to separate the original data, which have two features GPA and GRE, into two datasets $GPA2030$ and $GRE2030$, each having only one feature GPA or GRE, respectively.

   (a) We assume the two classes C_0 (not admitted) and C_1 (admitted) follow a Bernoulli distribution and we model the class conditional distributions as Gaussian distributions $\mathcal{N}(\mu_0, \sigma^2)$ and $\mathcal{N}(\mu_1, \sigma^2)$, respectively. When modeling with the same variance, this is also know as a Linear Discriminant Analysis (LDA) model. Find the maximum likelihood estimate of the parameters $P(y = 0)$ (parameter for the Bernoulli distribution), μ_0, μ_1 and σ^2 for datasets $GPA2030$ and $GRE2030$. Report your learned parameters for the two datasets.

   (b) The decision boundary for the GDA model is when the joint PDFs of the two classes are equal. When modeling with the same variance, the decision rule in the scalar case is of the form $x \lessgtr b$. For the two datasets, find and report the decision boundaries b. Also find and report the classification accuracy based on your decision boundary.

   (c) Visualization. Plot the following on the same plot for each of the datasets:
   - Data points from different classes in different colors on the x axis.
   - Joint PDFs (Gaussian density functions weighted by the class prior) for the two classes.
   - The decision boundary (a vertical line).

   Note: as a sanity check, the decision boundary should pass through the intersection of the joint PDFs.

   (d) Now we assume the two classes C_0 and C_1 follow a Bernoulli distribution and we model the class conditional distributions as Gaussian distributions $\mathcal{N}(\mu_0, \sigma_0^2)$ and $\mathcal{N}(\mu_1, \sigma_1^2)$, respectively. When modeling with different variances, this is also know as a Quadratic Discriminant Analysis (QDA) model. Find the maximum likelihood estimate of the parameters $P(y = 0)$ (parameter for the Bernoulli distribution), μ_0, μ_1, σ_0^2 and σ_1^2 for datasets $GPA2030$ and $GRE2030$. Report your learned parameters for the two datasets.

   (e) The decision boundary for the GDA model is when the joint PDFs of the two classes are equal. When modeling with different variances, the decision rule in the scalar case is of the form $ax^2 + bx + c \lessgtr 0$. For the two datasets, find and report the two roots of $ax^2 + bx + c = 0$. Also find and report the classification accuracy based on your decision rule.

   (f) Visualization. Plot the following on the same plot for each of the datasets:
   - Data points from different classes in different colors on the x axis.
   - Joint PDFs (Gaussian density functions weighted by the class prior) for the two classes.

- The line $ax^2 + bx + c$. Note that one of the roots of $ax^2 + bx + c = 0$ may not be within the range from 0 to 4. You don't have to show this root on your plot.