ENGR 421/DASC 521: Introduction to Machine Learning
Fall 2021 Final – Solution Key

Question 1:
We have two classes that are assumed to have one-dimensional Gaussian distributions with different means and variances: $p(x|y=1) \sim N(\mu_1, \sigma^2)$ and $p(x|y=2) \sim N(\mu_2, 4\sigma^2)$. Derive the position(s) of the intersection of the two posterior probabilities.

We would like to find x that satisfy $P(y=1|x) = P(y=2|x)$.

$$\begin{aligned} p(x|y=1)P(y=1) &= p(x|y=2)P(y=2) \\ \log p(x|y=1) + \log P(y=1) &= \log p(x|y=2) + \log P(y=2) \\ -\frac{\log(2\pi\sigma^2)}{2} - \frac{(x-\mu_1)^2}{2\sigma^2} + \log P(y=1) &= -\frac{\log(8\pi\sigma^2)}{2} - \frac{(x-\mu_2)^2}{8\sigma^2} + \log P(y=2) \end{aligned}$$

$$-\frac{3}{8\sigma^2}x^2 + \frac{4\mu_1 - \mu_2}{4\sigma^2}x - \frac{(2\mu_1 - \mu_2)(2\mu_1 + \mu_2)}{8\sigma^2} + \log\frac{P(y=1)}{P(y=2)} + \log 2 = 0$$

Question 2:
Consider a data set of N data points, in which each data point has one real-valued positive input x_i and the corresponding real-valued output y_i, i.e., $\{(x_i, y_i)\}_{i=1}^N$. We use the following model to fit the data, which has an unknown parameter w (the variance is known in advance and is set to 1).

$$p(y_i|x_i) \sim N(\log(wx_i), 1) \quad \forall i$$

(a) Describe a maximum likelihood approach to infer w and write down the log-likelihood objective for this problem.

(b) Find the maximum likelihood solution for w.

(a) By assuming the data points are independent from each other, we can write down the likelihood function as follows:

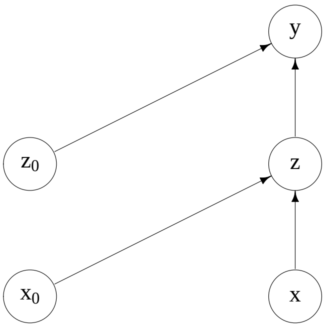
$$\begin{aligned} \text{likelihood} &= \prod_{i=1}^N p(y_i|x_i) = \prod_{i=1}^N N(\log(wx_i), 1) \\ &= \prod_{i=1}^N \frac{1}{\sqrt{2\pi(1)^2}} \exp\left(-\frac{(y_i - \log(wx_i))^2}{2(1)^2}\right) \end{aligned}$$

$$\text{log-likelihood} = \sum_{i=1}^N \left(-(1/2)\log(2\pi) - (1/2)(y_i - \log(wx_i))^2\right)$$

(b) To maximize log-likelihood, we need to minimize $\sum_{i=1}^{N}(1/2)(y_i - \log(wx_i))^2$ with respect to w.

$$\frac{\partial \sum_{i=1}^{N}(1/2)(y_i - \log(wx_i))^2}{\partial w} = \sum_{i=1}^{N} \frac{\partial (1/2)(y_i - \log(wx_i))^2}{\partial w}$$
$$= \sum_{i=1}^{N}(2(1/2)(y_i - \log(wx_i)))(-(x_i/(wx_i))) = 0$$
$$\sum_{i=1}^{N}(y_i - (\log(w^\star) + \log(x_i))) = 0 \Rightarrow N\log(w^\star) = \sum_{i=1}^{N}(y_i - \log(x_i))$$
$$w^\star = \exp\left(\left[\sum_{i=1}^{N}(y_i - \log(x_i))\right]/N\right)$$

Question 3:
Given a multilayer perceptron with one input, one tanh hidden unit, and one sigmoid output unit, derive the weight update equations to minimize the cross-entropy using gradient-descent.



$$\frac{\partial \tanh(a)}{\partial a} = (1 - \tanh(a))^2$$
$$\frac{\partial \mathrm{sigmoid}(a)}{\partial a} = \mathrm{sigmoid}(a)(1 - \mathrm{sigmoid}(a))$$
$$z_i = \tanh(wx_i + w_0)$$
$$\widehat{y}_i = \mathrm{sigmoid}(vz_i + v_0)$$
$$\mathrm{Error}_i = -y_i \log \widehat{y}_i - (1 - y_i)\log(1 - \widehat{y}_i)$$

$$\Delta v = -\eta \frac{\partial \mathrm{Error}_i}{\partial \widehat{y}_i}\frac{\partial \widehat{y}_i}{\partial v}$$
$$= \eta(y_i - \widehat{y}_i)z_i$$
$$\Delta v_0 = -\eta \frac{\partial \mathrm{Error}_i}{\partial \widehat{y}_i}\frac{\partial \widehat{y}_i}{\partial v_0}$$
$$= \eta(y_i - \widehat{y}_i)$$
$$\Delta w = -\eta \frac{\partial \mathrm{Error}_i}{\partial \widehat{y}_i}\frac{\partial \widehat{y}_i}{\partial z_i}\frac{\partial z_i}{\partial w}$$
$$= \eta(y_i - \widehat{y}_i)v(1 - z_i)^2 x_i$$
$$\Delta w_0 = -\eta \frac{\partial \mathrm{Error}_i}{\partial \widehat{y}_i}\frac{\partial \widehat{y}_i}{\partial z_i}\frac{\partial z_i}{\partial w_0}$$
$$= \eta(y_i - \widehat{y}_i)v(1 - z_i)^2$$

Question 4:
We know that an $N \times N$ symmetric real matrix $\mathbf{K}$ is said to be positive semidefinite if $\boldsymbol{a}^\top \mathbf{K}\boldsymbol{a} \geq 0$ for all $\boldsymbol{a}$ in $\mathbb{R}^N$.

(a) Show that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$ produces a positive semidefinite kernel matrix on a given set of N data points, i.e., $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$.

(b) Show that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \dfrac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_j\|_2}$ produces a positive semidefinite kernel matrix on a given set of N data points, i.e., $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$.

(a) $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$ produces the following kernel matrix:

$$\mathbf{K} = \begin{bmatrix} \boldsymbol{x}_1^\top \boldsymbol{x}_1 & \boldsymbol{x}_1^\top \boldsymbol{x}_2 & \dots & \boldsymbol{x}_1^\top \boldsymbol{x}_N \\ \boldsymbol{x}_2^\top \boldsymbol{x}_1 & \boldsymbol{x}_2^\top \boldsymbol{x}_2 & \dots & \boldsymbol{x}_2^\top \boldsymbol{x}_N \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_N^\top \boldsymbol{x}_1 & \boldsymbol{x}_N^\top \boldsymbol{x}_2 & \dots & \boldsymbol{x}_N^\top \boldsymbol{x}_N \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{bmatrix}^\top}_{\mathbf{X}^\top}$$

$$\begin{aligned} \boldsymbol{a}^\top \mathbf{K} \boldsymbol{a} &\geq 0 \\ \boldsymbol{a}^\top \mathbf{X} \mathbf{X}^\top \boldsymbol{a} &\geq 0 \\ (\mathbf{X}^\top \boldsymbol{a})^\top \mathbf{X}^\top \boldsymbol{a} &\geq 0 \\ \|\mathbf{X}^\top \boldsymbol{a}\|_2^2 &\geq 0 \end{aligned}$$

(b) $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \dfrac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{\|\boldsymbol{x}_i\|_2 \|\boldsymbol{x}_j\|_2}$ produces the following kernel matrix:

$$\mathbf{K} = \begin{bmatrix} \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_1}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_1\|_2} & \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2} & \dots & \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_N}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_N\|_2} \\ \frac{\boldsymbol{x}_2^\top \boldsymbol{x}_1}{\|\boldsymbol{x}_2\|_2 \|\boldsymbol{x}_1\|_2} & \frac{\boldsymbol{x}_2^\top \boldsymbol{x}_2}{\|\boldsymbol{x}_2\|_2 \|\boldsymbol{x}_2\|_2} & \dots & \frac{\boldsymbol{x}_2^\top \boldsymbol{x}_N}{\|\boldsymbol{x}_2\|_2 \|\boldsymbol{x}_N\|_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\boldsymbol{x}_N^\top \boldsymbol{x}_1}{\|\boldsymbol{x}_N\|_2 \|\boldsymbol{x}_1\|_2} & \frac{\boldsymbol{x}_N^\top \boldsymbol{x}_2}{\|\boldsymbol{x}_N\|_2 \|\boldsymbol{x}_2\|_2} & \dots & \frac{\boldsymbol{x}_N^\top \boldsymbol{x}_N}{\|\boldsymbol{x}_N\|_2 \|\boldsymbol{x}_N\|_2} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\boldsymbol{x}_1^\top}{\|\boldsymbol{x}_1\|_2} \\ \frac{\boldsymbol{x}_2^\top}{\|\boldsymbol{x}_2\|_2} \\ \vdots \\ \frac{\boldsymbol{x}_N^\top}{\|\boldsymbol{x}_N\|_2} \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \frac{\boldsymbol{x}_1^\top}{\|\boldsymbol{x}_1\|_2} \\ \frac{\boldsymbol{x}_2^\top}{\|\boldsymbol{x}_2\|_2} \\ \vdots \\ \frac{\boldsymbol{x}_N^\top}{\|\boldsymbol{x}_N\|_2} \end{bmatrix}^\top}_{\mathbf{U}^\top}$$

$$\begin{aligned} \boldsymbol{a}^\top \mathbf{K} \boldsymbol{a} &\geq 0 \\ \boldsymbol{a}^\top \mathbf{U} \mathbf{U}^\top \boldsymbol{a} &\geq 0 \\ (\mathbf{U}^\top \boldsymbol{a})^\top \mathbf{U}^\top \boldsymbol{a} &\geq 0 \\ \|\mathbf{U}^\top \boldsymbol{a}\|_2^2 &\geq 0 \end{aligned}$$

Question 5:
The XOR problem is given by:

$\boldsymbol{x}_i$	x_{i1}	x_{i2}	y_i
$\boldsymbol{x}_1$	-1	-1	-1
$\boldsymbol{x}_2$	$+1$	-1	$+1$
$\boldsymbol{x}_3$	-1	$+1$	$+1$
$\boldsymbol{x}_4$	$+1$	$+1$	-1

If we use the following mapping function $\Phi(\cdot)$ for the support vector machine formulation, can we solve this binary classification problem successfully? Justify your answer.

$$\Phi(\boldsymbol{x}_i) = \begin{bmatrix} x_{i1}^2 - x_{i2}^2 \\ x_{i1}x_{i2} \\ x_{i1}^2 + x_{i2}^2 \end{bmatrix}$$

The mapping function produces the following representations: $\Phi(\boldsymbol{x}_1) = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}^\top$, $\Phi(\boldsymbol{x}_2) = \begin{bmatrix} 0 & -1 & 2 \end{bmatrix}^\top$, $\Phi(\boldsymbol{x}_3) = \begin{bmatrix} 0 & -1 & 2 \end{bmatrix}^\top$, and $\Phi(\boldsymbol{x}_4) = \begin{bmatrix} 0 & 1 & 2 \end{bmatrix}^\top$.

The kernel matrix can be constructed as follows:

$$\mathbf{K} = \begin{bmatrix} 5 & 3 & 3 & 5 \\ 3 & 5 & 5 & 3 \\ 3 & 5 & 5 & 3 \\ 5 & 3 & 3 & 5 \end{bmatrix}$$

$$\begin{aligned} \text{maximize} \quad & J = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^\top \begin{bmatrix} 5 & -3 & -3 & 5 \\ -3 & 5 & 5 & -3 \\ -3 & 5 & 5 & -3 \\ 5 & -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \\ \text{with respect to} \quad & \alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4 \\ \text{subject to} \quad & -\alpha_1 + \alpha_2 + \alpha_3 - \alpha_4 = 0 \\ & \alpha_i \geq 0 \quad i \in \{1, 2, 3, 4\} \end{aligned}$$

$$\begin{aligned} \frac{\partial J}{\partial \alpha_1} = \frac{\partial J}{\partial \alpha_4} &= 1 - 5\alpha_1 + 3\alpha_2 + 3\alpha_3 - 5\alpha_4 = 0 \\ \frac{\partial J}{\partial \alpha_2} = \frac{\partial J}{\partial \alpha_3} &= 1 + 3\alpha_1 - 5\alpha_2 - 5\alpha_3 + 3\alpha_4 = 0 \end{aligned}$$

For example, $\alpha_1^\star = 1/2$, $\alpha_2^\star = 1/2$, $\alpha_3^\star = 0$, and $\alpha_4^\star = 0$ is one of the optimum solutions.

Question 6:
Recall the error function for k-means clustering with K clusters, data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, and centers $\widehat{\boldsymbol{\mu}}_1, \ldots, \widehat{\boldsymbol{\mu}}_K$:

$$E = \sum_{i=1}^{N} \sum_{k=1}^{K} b_{ik} \|\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k\|_2^2$$

where b_{ik} is equal to 1 if data point $\boldsymbol{x}_i$ is closer to center $\widehat{\boldsymbol{\mu}}_k$ than to any other center and to 0 otherwise.

(a) Instead of updating $\{\widehat{\boldsymbol{\mu}}_k\}_{k=1}^{K}$ by computing the means, let us minimize E with batch gradient descent while holding $\{b_{ik}\}_{i=1,k=1}^{N,K}$ fixed. Derive the update formula for $\widehat{\boldsymbol{\mu}}_1$ with learning rate η.

(b) Derive the update formula for $\widehat{\boldsymbol{\mu}}_1$ with gradient descent on a single data point $\boldsymbol{x}_i$. Use learning rate η.

(c) Recall that in the update step of the standard algorithm, we assign each cluster center to the mean of the data points closest to that center. It turns out that a particular choice of the learning rate η, which may be different for each cluster, makes the two algorithms (batch gradient descent and the standard k-means algorithm) have identical update steps. Let us focus on the update for the first cluster, with center $\widehat{\boldsymbol{\mu}}_1$. Calculate the value of η so that both algorithms perform the same update for $\widehat{\boldsymbol{\mu}}_1$.

(a)

$$\frac{\partial E}{\partial \widehat{\boldsymbol{\mu}}_1} = \frac{\partial \left(\sum_{i=1}^{N} b_{i1} \|\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1\|_2^2 + \text{constant} \right)}{\partial \widehat{\boldsymbol{\mu}}_1} = -2 \sum_{i=1}^{N} b_{i1} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)$$

Therefore the update formula is $\widehat{\boldsymbol{\mu}}_1^{(t+1)} = \widehat{\boldsymbol{\mu}}_1^{(t)} + 2\eta \sum_{i=1}^{N} b_{i1} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1^{(t)})$

(b)

$$\frac{\partial E_i}{\partial \widehat{\boldsymbol{\mu}}_1} = \frac{\partial (b_{i1} \|\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1\|_2^2 + \text{constant})}{\partial \widehat{\boldsymbol{\mu}}_1} = -2 b_{i1} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1)$$

Therefore the update formula is $\widehat{\boldsymbol{\mu}}_1^{(t+1)} = \widehat{\boldsymbol{\mu}}_1^{(t)} + 2\eta b_{i1} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1^{(t)})$

(c) In the standard k-means algorithm, we assign $\widehat{\boldsymbol{\mu}}_1^{(t+1)} = \dfrac{\sum_{i=1}^{N} b_{i1} \boldsymbol{x}_i}{\sum_{i=1}^{N} b_{i1}}$.

$$\frac{\sum_{i=1}^{N} b_{i1} \boldsymbol{x}_i}{\sum_{i=1}^{N} b_{i1}} = \widehat{\boldsymbol{\mu}}_1^{(t)} + 2\eta \sum_{i=1}^{N} b_{i1} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_1^{(t)}) \Rightarrow \eta = \frac{1}{2 \sum_{i=1}^{N} b_{i1}}$$

Question 7:

One may be concerned that the randomness introduced in random forests may cause trouble, for instance, some features or samples may not be considered at all.

(a) Consider N training samples in a feature space of D dimensions. Consider building a random forest with T binary trees, each having exactly H internal nodes. Let F be the number of features randomly selected at each node. In order to simplify our calculations, we will let $F = 1$. For this setting, compute the probability that a certain feature (say, the first feature) is never considered for splitting.

(b) Now let us investigate the concern regarding the random selection of the samples. Suppose each tree employs N bootstrapped training samples. Compute the probability that a particular sample (say, the first sample) is never considered in any of the trees.

(c) Compute the values of the probabilities you obtained in the previous two parts for the case when there are $N = 2$ training samples, $D = 2$ dimensions, $T = 10$ trees of depth $H = 4$. What conclusions can you draw from your answer with regard to the concern mentioned at the beginning of the question?

(a) The probability that it is not considered for splitting in a particular node of a particular tree is $(1 - 1/D)$. The subsampling of $F = 1$ features at each node is independent of all others. There are a total of TH nodes and hence the final answer is $(1 - 1/D)^{TH}$.

(b) The probability that it is not considered in one of the trees is $(1 - 1/N)^N$. Since the choice for every tree is independent, the probability that it is not considered in any of the trees is $(1 - 1/N)^{NT}$.

(c) $(1/2)^{40}$ and $(1/2)^{20}$. It is quite unlikely that a feature or a sample will be missed.

Question 8:
It is suggested to run the k-means clustering algorithm multiple times with different initializations. Explain the reasoning behind this suggestion.

The k-means clustering algorithm is heavily affected by the initial configuration. That is why it is always a good idea to run the algorithm multiple times and pick the best solution.

Question 9:
Let a configuration of the k-means clustering algorithm correspond to the k way partition (on the set of instances to be clustered) generated by the clustering at the end of each iteration. Is it possible for the k-means algorithm to revisit a configuration? Justify your answer.

It is guaranteed that the objective function value of the k-means clustering algorithm is monotonically decreasing during successive iterations. That is why it is not possible to revisit a configuration once you move to another but a better configuration.

Question 10:
What are the distance and linkage functions used in hierarchical clustering algorithms? How do they differ in their roles?

Distance function: Distance measure between two instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Linkage function: Distance measure between two groups $\mathcal{G}_A$ and $\mathcal{G}_B$.

Question 11:
What is an unstable learner? Why does bagging (i.e., bootstrap aggregating) rely on having an unstable learner as the base classifier?

A learning algorithm is unstable if small changes in the training set causes a large difference in the generated learner.

Bagging trains base-learners on similar data sets and, in order to generate diverse learners on these data sets, we need to have unstable learners.

Question 12:
What are the advantages of K-fold cross-validation and 5×2 cross-validation on each other?

K-fold cross-validation:

+ Training set size is large

− Overlap between training sets is large

5×2 cross-validation:

- $+$ Overlap between training sets is small
- $-$ Training set size is small

Question 13:
For classification algorithms, there are different performance metrics such as classification accuracy, area under the ROC curve, precision, recall, etc. Explain when it is not a good idea to use the classification accuracy.

For heavily imbalanced data sets, it is not a good idea to use the classification accuracy as the performance metric since returning the majority class for all test samples will give a very high classification accuracy.