CM146: Introduction to Machine Learning **Fall 2020**

Final Exam

Dec 18, 2020

- **Read the instructions below prior to starting the exam!**
- This is an open book exam – you can use course materials posted at CCLE as your reference. Please do not access any other material during the exam. Discussion is strictly prohibited.
- This exam booklet contains **seven** problems.
- You have 180 minutes to earn a total of 100 points.
- You can ask *private* question in Piazza, but only clarification questions will be answered. If there is any issue of the exam, we will make announcement before before Friday 12/18 2:30PM. If you have a doubt about the question, feel free to write down your explanations. The grading will only based on the solutions written in the exam booklet.
- If you type the answers, it is okay to use latex convention, e.g., x_1, \gamma
- **The deadline to upload your exam to Gradescope is *Dec 19 11:30am.* No late submissions will be accepted, so make sure to submit well before the deadline!**

Good Luck!

Name and ID:

1 Short Answer Questions and Multiple Choices [18 pts]

(a) **(3 pts)** Z is a summer intern working on *spam classification* in your company. The dataset consists of 10 million non-spam emails (class 0) and 10 thousand spam emails (class 1). Z considers the following steps of conducting experiments:

- Step 1: Shuffle the dataset and split it into train, validation, and test sets.
- Step 2: Train logistic regression models on the train set with different hyper-parameters.
- Step 3: Identify the best hyper-parameter using the validation set and report the results on the test set in accuracy.

Do you agree with the above experimental setup? If No, what is the major issue? Provide your suggestions in one or two sentences.

Solution: No, when the class labels are extremely unbalanced, accuracy is not a good measurement. F1 score should be used as the evaluation metric instead.

(b) **(3 pts)** Continue 1(a). Z decides to use the scikit-learn library for data standardization before training the model. Z sends you the following code snippet for code review.

```
...
# X_train, X_val and X_test contain train, val and test data
scaler = preprocessing.StandardScaler()
X_train_std = scaler.fit_transform(X_train)
X_val_std = scaler.fit_transform(X_val)
X_test_std = scaler.fit_transform(X_test)
# Z uses X_train_std, X_val_std and X_test_std for train, validation and test
...
```

Would you approve Z's code? If No, please show how to correct the code.

Hint: the following scikit-learn documentation might be useful:

- *fit()* - Compute the mean and std to be used for later scaling.
- *transform()* - Perform standardization by centering and scaling. Should be called after *fit()*.
- *fit_transform()* - Fit the data, then transform it.

Solution: No, the same transformation should be applied to train/validation/test sets. One potential modification is computed the transformation only from train data and applied to validation and test sets.

```
...
# X_train, X_val and X_test contain train, val and test data
scaler = preprocessing.StandardScaler()
X_train_std = scaler.fit_transform(X_train)
X_val_std = scaler.transform(X_val)
X_test_std = scaler.transform(X_test)
# Z uses X_train_std, X_val_std and X_test_std for train, validation and test
...
```

(c) **(2 pts)** Which of the following statement(s) about ID3 algorithm are correct? Select all of them.

☐ (A) The ID3 algorithm always finds the optimal decision tree, i.e., the decision tree with the minimal depth that can classify all training instances.

☐ (B) The ID3 algorithm can be only used in binary classification problems.

☐ (C) ID3 algorithm can be used to find a non-linear classifier.

☐ (D) Decision trees can be implemented as a set of if-then-else statements.

Solution: C, D

(d) **(2 pts)** Which of the following statement(s) related to multi-class classification are correct? Select all of them.

☐ (A) One-vs-One strategy decomposes a multi-class classification problem into several binary classification problems.

☐ (B) For the same multi-class classification problem, it is possible that some of the corresponding binary classification problems are not linearly separable when using the One-against-All strategy, while the binary classification problems are all linearly separable when using the One-vs-One strategy.

☐ (C) Imbalanced training set size is a common issue with One-vs-One strategy

☐ (D) One-vs-One strategy requires to train more binary classifiers than the One-against-All strategy when the number of classes is 5.

Solution: ABD

(e) **(3 pts)** Consider a linear regression model $f : y = 0.5x + 1$. Given a set of data points $D = \{(x, y)\} = (1.0, 1.6), (1.5, 1.5), (3.0, 2.4)$. What is the mean squared error of the model f on D? Write down your final answer.

Ans: _____

Solution: MSE $= \frac{0.1^2+0.25^2+0.1^2}{3} = 0.0275.$

(f) **(5 pts)** Consider $x \in \mathbb{R}$ and assume we have six data points $x_1 = 2$, $x_2 = 3$, $x_3 = 7$, $x_4 = 12$, $x_5 = 15$, $x_6 = 18$. In the following, we are going to apply K-means algorithm with $K = 2$ on the following six points. The initial centers $c_1 = 13$, $c_2 = 16$. Complete the following table.

Initialization:	$c_1 = 13$	$c_2 = 16$
Step 1:	Cluster 1: x_1, x_2, x_3, x_4	Cluster 2: x_5, x_6
Step 2:	$c_1 =$_____	$c_2 =$_____
Step 3:	Cluster 1: _____	Cluster 2: _____
Step 4:	$c_1 =$ _____	$c_2 =$ _____
Step 5:	Cluster 1: _____	Cluster 2: _____
Step 6:	$c_1 =$ _____	$c_2 =$ _____

Solution:

Initialization:	$c_1 = 13$	$c_2 = 16$
Step 1:	Cluster 1: x_1, x_2, x_3, x_4	Cluster 2: x_5, x_6
Step 2:	$c_1 = 6$	$c_2 = 16.5$
Step 3:	Cluster 1: x_1, x_2, x_3	Cluster 2: x_4, x_5, x_6
Step 4:	$c_1 = 4$	$c_2 = 15$
Step 5:	Cluster 1: x_1, x_2, x_3	Cluster 2: x_4, x_5, x_6
Step 6:	$c_1 = 4$	$c_2 = 15$

2 Maximum Likelihood Estimation (MLE) [9 pts]

In this question, we will explore how can we predict the number of passengers waiting in LAX at a specific time. Flights could be delayed for various reasons including weather, humidity, and heavy traffic on the airport runways. Suppose that we have collected all those related features and converted them to numerical variables: x_n. We also managed to accurately count the number (non-negative integer) of passengers y_n waiting in LAX on day $\boldsymbol{n}$ from Jan 1 2017 to Dec 31 2017 to construct a training data set $\{(x_n, y_n)\}_{n=1}^N$, where $x_n \in \mathbb{R}^M$, $y_n \in \{0, 1, 2, \ldots\}$. We assume the mapping between target variable y and the input feature vector x can be modeled by a Poisson distribution with parameter θ:

$$P(Y = y|X = x; \theta) = \frac{\lambda^y}{y!} \cdot e^{-\lambda} \quad \text{where } \lambda = e^{\theta^T x}.$$

Use the knowledge you learned about MLE to answer the following questions.

(a) **(3 pts)** What is the likelihood function for one specific training example (x_n, y_n)?

- ☐ (A) $P(y_n|x_n, \theta) = y_n(\theta^T x_n) - \theta^T x_n + constant$
- ☐ (B) $P(y_n|x_n, \theta) = \frac{1}{y_n!} \cdot e^{y_n \theta^T x_n - \theta^T x_n}$
- ☐ (C) $P(y_n|x_n, \theta) = \frac{1}{y_n!} \cdot e^{y_n \theta^T x_n} \cdot e^{-e^{\theta^T x_n}}$
- ☐ (D) $P(y_n|x_n, \theta) = \frac{1}{y_n!} \cdot (e^{\theta^T x_n})^{y_n} \cdot e^{-\theta}$
- ☐ (E) $P(y_n|x_n, \theta) = (\theta^T x_n)^{y_n} - \theta^T x_n + constant$

Solution: C

(b) **(3 pts)** If we assume the training examples are drawn i.i.d. from the underlying data distribution, what is the log likelihood of the training set $\{(x_n, y_n)\}_{n=1}^N$?

- ☐ (A) $\mathcal{LL}(\theta) = constant + \sum_{n=1}^N y_n e^{\theta^T x_n} - \theta$
- ☐ (B) $\mathcal{LL}(\theta) = constant \cdot \prod_{n=1}^N y_n \theta^T x_n - \theta^T x_n$
- ☐ (C) $\mathcal{LL}(\theta) = constant + \sum_{n=1}^N (\theta^T x_n)^{y_n} - \theta^T x_n$
- ☐ (D) $\mathcal{LL}(\theta) = constant + \sum_{n=1}^N y_n \theta^T x_n - \theta^T x_n$
- ☐ (E) $\mathcal{LL}(\theta) = constant + \sum_{n=1}^N y_n \theta^T x_n - e^{\theta^T x_n}$

Solution: E

(c) **(3 pts)** Considering the above Poisson regression model, which of the following statement is correct? Hint: $e^{\theta^T x}$ is a convex function w.r.t θ.

☐ (A) $\mathcal{LL}(\theta)$ is a convex function.

☐ (B) $\mathcal{LL}(\theta)$ is a concave function.

☐ (C) $\mathcal{LL}(\theta)$ is not a convex nor a concave function.

☐ (D) $\mathcal{LL}(\theta)$ is both a convex nor a concave function.

Solution: B, Because $e^{\theta^T x}$ is convex, $-e^{\theta^T x}$ is a concave function. a concave function adds a linear function is still a concave function.

3 Gaussian Mixture Model [14 pts]

In the following, we consider a **hard-assignment** GMM. The hard-assignment GMM is similar to the soft-assignment GMM we learned in the class. However, instead of having $\gamma_{nk} \in [0, 1]$, in the hard-assignment GMM, γ_{nk} is a Boolean variable that is set to 1 if and only if the data point n belongs to cluster k. Formally, we consider the data set consists of N i.i.d. data points $\{x_n \in \mathbb{R}\}_{n=1}^N$ and our goal is cluster into K groups using K Gaussians $\mathcal{N}(x_n; \mu_k, \sigma_k^2), k = 1, 2, \ldots, K$. The prior probability of the cluster k is ω_k. We use $\theta = \{\omega_k, \mu_k, \sigma_k\}_{k=1}^K$ to represent all model parameters and z_n is a random variable to represent the cluster assignment of the n-th data point.

Probability density function for normal distribution:

$$\mathbf{P}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

(a) **(2 pts)** What is the value of $\sum_{k=1}^K \gamma_{nk}$?

$$\sum_{k=1}^{K} \gamma_{nk} = \underline{\qquad\qquad\qquad\qquad}$$

Solution: 1

(b) **(4 pts)** GMM is an iterative algorithm. We alternatively update γ_{nk} and θ. Given a fixed γ_{nk}, we derive the optimal θ in the following. Please complete the following derivation by filling the blanks:

$$\begin{aligned}
\mathcal{LL}(\theta) &= \sum_{n=1}^{N} \log P(x_n, z_n|\theta) \\
&= \sum_{n=1}^{N} \log \prod_{k=1}^{K} P(x_n, z_n|\theta)^{\gamma_{nk}} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \underline{\quad \text{blank\#1} \quad \textbf{Solution :} \gamma_{nk}} \log P(x_n, z_n|\theta) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \{ \underline{\qquad\qquad \text{blank\#2} \qquad\qquad \textbf{Solution :} \log P(x_n|z_n, \theta)} + \log P(z_n|\theta) \} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log \mathcal{N}(x_n; \mu_k, \sigma_k^2) + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log w_k
\end{aligned}$$

(c) **(4 pts)** Let $\mu_l^* = \arg\max_{\mu_l} \mathcal{LL}(\theta)$ is the optimal solution of maximizing $\mathcal{LL}(\theta)$ with respect to μ_l. Which of the following equations are true? Select all of them.

☐ (A) $\mu_l^* = \arg\max_{\mu_l} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \cdot \log \mathcal{N}(x_n; \mu_k, \sigma_k^2)$

☐ (B) $\mu_l^* = \arg\min_{\mu_l} \sum_{n=1}^{N} \gamma_{nl} \cdot \log \sigma_l^2 + \frac{\gamma_{nl} \cdot (x_n - \mu_l)^2}{2} + \gamma_{nl} \cdot \log w_l$

☐ (C) $\mu_l^* = \arg\max_{\mu_l} \sum_{n=1}^{N} \gamma_{nl} \cdot \log \mathcal{N}(x_n; \mu_l, \sigma_l^2)$

☐ (D) $\mu_l^* = \arg\min_{\mu_l} \sum_{n=1}^{N} \gamma_{nl} \cdot (x_n - \mu_l)^2$

Solution: ABCD

$$\begin{aligned}
\boldsymbol{\mu_l}^* = \arg\max_{\boldsymbol{\mu_l}} \mathcal{LL}(\theta) &= \arg\max_{\boldsymbol{\mu_l}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nl} \cdot \log \{\mathbb{N}(x_n; \boldsymbol{\mu_l}, \boldsymbol{\sigma_l^2}) \cdot \boldsymbol{w_l}\} \\
&= \arg\max_{\boldsymbol{\mu_l}} \sum_{n=1}^{N} \gamma_{nl} \cdot \log \mathbb{N}(x_n; \boldsymbol{\mu_l}, \boldsymbol{\sigma_l^2}) \\
&= \arg\max_{\boldsymbol{\mu_l}} \sum_{n=1}^{N} \gamma_{nl} \cdot \{-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma_l^2 - \frac{1}{2}\frac{(x_n - \mu_l)^2}{\sigma_l^2}\} \\
&= \arg\min_{\boldsymbol{\mu_l}} \sum_{n=1}^{N} \gamma_{nl} \cdot (x_n - \mu_l)^2
\end{aligned}$$

The option B is also correct as the fist and third terms are constant and the factor of $\frac{1}{2\sigma_l}$ is a constant to μ_l as well.

find the second derivative with respect to $\mu_l = 2 \cdot \sum_{n=1}^{n=N} \gamma_{nl} \geq 0$ Thus the function is convex and we can find the MLE by setting the gradient to 0

$$-2 \cdot \sum_{n=1}^{N} \gamma_{nl}(x_n - \mu_l^*) = 0$$

$$\sum_{n=1}^{N} \gamma_{nl}\, x_n = \sum_{n=1}^{N} \gamma_{nl}\, \mu_l^*$$

$$\mu_l^* = \frac{\sum_{n=1}^{N} \gamma_{nl} x_n}{\sum_{n=1}^{N} \gamma_{nl}}$$

Notice that this is the same update for Kmeans algorithm. So you can think of KMeans as a Hard SVM

(d) **(4 pts)** To update γ_{nk}, we assign each data point to the cluster with largest $P(z_n \mid x_n; \theta)$ (i.e., $z_n^* = \arg\max_k P(z_n = k \mid x_n; \theta)$. Which of the following statements are true? Select all of them.

☐ (A) $z_n^* = \arg\max_k \mathcal{N}(x_n; \mu_k, \sigma_k^2)$

☐ (B) $z_n^* = \arg\max_k \omega_k \cdot \mathcal{N}(x_n; \mu_k, \sigma_k^2)$

☐ (C) $z_n^* = \arg\max_k \log \omega_k + \log \mathcal{N}(x_n; \mu_k, \sigma_k^2)$

☐ (D) $z_n^* = \arg\max_k \log \omega_k$

Solution: BC

4 Expectation Maximization (EM) [13 pts]

Suppose we have a simulator that simultaneously exhibits 2 field forces f_1, f_2 on a robot. Both field force generators *independently* generate field forces E or B with probabilities ϕ_E, ϕ_B, respectively ($\phi_E + \phi_B = 1$). The robot is exposed to two combination forces at each step. We do not observe the field forces but we observe the robot's movement moving either forward (**fwd**) or backward (**bwd**). Specifically, the robot moves forward when the combination of the two field forces is either EE, EB, or BE. On the other hand the robot moves backwards if the field force combination is BB.

Forces (F)	$P(F)$	Robot Movement (M)
EE	ϕ_E^2	fwd
EB	$\phi_E\phi_B$	fwd
BE	$\phi_B\phi_E$	fwd
BB	ϕ_B^2	bwd

We conduct N experiment trials, and we observe that the robot moves forward n_{fwd} times and moves backward n_{bwd} times ($n_{fwd} + n_{bwd} = N$). Our objective is to determine ϕ_E and ϕ_B using EM algorithm based on the robot's movements in these N experiment trials.

As a reminder, EM is an iterative process. In the E-Step, we estimate the numbers of expected force combinations, n_{EE}, n_{BB}, n_{BE} n_{EB} in the N trials ($\mathrm{n}_{EE} + \mathrm{n}_{BE} + \mathrm{n}_{EB} + \mathrm{n}_{BB} = N$) based on ϕ_E and ϕ_B. In the M-Step, we estimate ϕ_E and ϕ_B based on n_{EE}, n_{BB}, n_{BE} n_{EB}.

(a) **(4 pts)** We first derive the E-Step. Suppose we are given initial estimates ϕ_E and ϕ_B, calculate the expected n_{EE}, $n_{EB} = n_{BE}$ and n_{BB} from the observed data.

(A) $n_{EE} = (n_{fwd} + n_{bwd})\phi_E, \quad n_{BB} = (n_{fwd} + n_{bwd})\phi_B, \quad n_{EB} = n_{BE} = (n_{fwd} + n_{bwd})\phi_B\phi_E$

(B) $n_{EE} = n_{fwd}\frac{\phi_E}{2\phi_E\phi_B+\phi_B^2}, \quad n_{BB} = n_{bwd} \quad n_{EB} = n_{BE} = (n_{fwd} + n_{bwd})\phi_B\phi_E$

(C) $n_{EE} = n_{fwd}\frac{\phi_E^2}{2\phi_E\phi_B+\phi_E^2}, \quad n_{BB} = n_{bwd}, \quad n_{EB} = n_{BE} = n_{fwd}\frac{\phi_E\phi_B}{2\phi_E\phi_B+\phi_E^2}$

(D) $n_{EE} = n_{fwd}\frac{\phi_E^2}{2\phi_E\phi_B+\phi_B^2}, \quad n_{BB} = n_{bwd}\frac{\phi_B^2}{\phi_B^2+2\phi_E\phi_B} \quad n_{EB} = n_{BE} = (n_{fwd} + n_{bwd})\frac{2\phi_E\phi_B}{2\phi_E\phi_B+\phi_E^2}$

Solution: C

(b) **(3 pts)** Next, we derive the M-Step. Assume if we know the expected numbers of force combinations are n_{EE}, n_{BB}, n_{BE} n_{EB}. Which of the following correspond to the likelihood function?

☐ (A) $\phi_E^{2n_{EE}+n_{EB}+n_{BE}}\phi_B^{2n_{BB}+n_{EB}+n_{BE}}$

☐ (B) $\phi_E^{n_{EE}+n_{EB}}\phi_B^{n_{BB}+n_{BE}}$

☐ (C) $\phi_E^{n_{EE}+n_{EB}+n_{BE}}\phi_B^{n_{BB}+n_{EB}+n_{BE}}$

☐ (D) $\phi_E^{n_{EE}+n_{EB}+n_{BE}+n_{BB}}\phi_B^{n_{EE}+n_{EB}+n_{BE}+n_{BB}}$

Solution:
A

(c) **(3 pts)** Continued the previous question, write down the M-Step for estimating ϕ_E and ϕ_B by maximizing the likelihood function.

☐ (A) $\phi_E = \frac{2n_{EE}+n_{BE}+n_{EB}}{2N}$ $\phi_B = \frac{2n_{BB}+n_{EB}+n_{BE}}{2N}$

☐ (B) $\phi_E = \frac{n_{EE}+2n_{EB}}{N}$ $\phi_B = \frac{n_{BB}+2n_{EB}}{N}$

☐ (C) $\phi_E = \frac{n_{EE}+2n_{EB}}{N}$ $\phi_B = \frac{n_{BB}}{N}$

☐ (D) $\phi_E = \frac{n_{EE}}{2N}$ $\phi_B = \frac{n_{BB}}{2N}$

Solution: A

(d) **(3 pts)** What are the properties of this EM algorithm? Select all of the true statements.

☐ (A) The EM algorithm is guaranteed to converge to the global optimum regardless of initialization.

☐ (B) The EM algorithm is only guaranteed to converge to a local optimum.

☐ (C) If we select a good initialization, it is possible that EM converges to the global optimum.

☐ (D) There is no way the EM algorithm can converge to the global optimum.

Solution: B, C

5 Learning Theory (22 pts)

Ring Classifier: In this problem, we consider data points in 2-dimensional space $x = (x_1, x_2) \in \mathbb{R}^2$. A ring classifier assigns the label 1 to a data point x if and only if x is inside a ring. Formally, given $t \leq r$, where $t, r \in \mathbb{R}$, a ring classifier $h_{(t,r)}$ labels data x by

$$h_{(t,r)}(x) = \begin{cases} 1, & \text{if } t \leq \|x\|_2 \leq r \\ 0, & \text{otherwise} \end{cases}.$$

In the following, we consider the ring hypothesis set

$$\mathcal{H}_{ring} = \{h_{(t,r)} : t \leq r\}.$$

We assume that the training dataset S_{train} consists of N examples x_i drawn i.i.d. from a distribution $\mathcal{D}$. The labels are provided by a target function $h^*_{(t^*,r^*)} \in H_{ring}$. In addition, we use S_p to denote the set of positive examples in S_{train} (i.e. they are inside the ring of $h^*_{(t^*,r^*)}$) and use S_n to denote the set of negative examples in S_{train}.

(a) **(4 pts)** Let $\mathcal{A}$ be an algorithm that learns to select a hypothesis $\mathcal{A}(S_{train}) \in \mathcal{H}_{ring}$ from the training dataset S_{train}, where $\mathcal{A}(S_{train})$ is the tightest ring enclosing all positive examples in S_{train}. Specifically, $\mathcal{A}(S_{train}) = h_{(t_a,r_a)}$, where

$$t_a = \min_{x \in S_p} \|x\|_2, \quad r_a = \max_{x \in S_p} \|x\|_2.$$

Show that $\mathcal{A}(S_{train})$ achieves zero traning error.

Solution: Since

$$t_a = \min_{x \in S_p} \|x\|_2, \quad r_a = \max_{x \in S_p} \|x\|_2,$$

we know all positive examples are in the ring of $h_{(t_a,r_a)}$ and they are correctly predicted. Next, from the definition of $h_{(t_a,r_a)}$, we know all the negative examples outside the ring of $h_{(t_a,r_a)}$ are correctly predicted. Finally, since the label of data point is generated by the target function $h^*_{(t^*,r^*)}$, we have $t^* \leq \|x\|_2 \leq r^*$ for all $x \in S_p$. That means $t^* \leq t_a \leq r_a \leq r^*$. Therefore, there is no negative examples in the ring of $h_{(t_a,r_a)}$. Combine all of above, we know all the training examples are correctly predicted and thus the training error is zero.

(b) **(4 pts)** We draw another set of samples S_{test} from D as the test set. Prove that $\mathcal{A}(S_{train})$ will not make any mistake on *negative examples* in S_{test}.

Solution: From (a), we know that $t^* \leq t_a \leq r_a \leq r^*$. For any negative testing example x, we have either $\|x\|_2 < t^* \leq t_a$ or $r_a \leq r^* < \|x\|_2$. Therefore, $\mathcal{A}(S_{train})$ will not make any mistake on negative examples. For those positive examples with $t^* < \|x\|_2 < t_a$ ot $r_a < \|x\|_2 < r^*$, $\mathcal{A}(S_{train})$ will make mistakes.

(c) **(4 pts)** We define the error as

$$\epsilon = \mathbf{P}_{x\sim\mathcal{D}}\Big(h^*_{(t^*,r^*)}(x) \neq h_{(t_a,r_a)}(x)\Big).$$

If we draw i.i.d. m examples from $\mathcal{D}$, what is the probability that $h_{(t_a,r_a)}$ makes no mistakes for all m examples?

☐ (A) $\left(\frac{\epsilon}{1-\epsilon}\right)^m$

☐ (B) $\left(\frac{1-\epsilon}{\epsilon}\right)^m$

☐ (C) ϵ^m

☐ (D) $(1-\epsilon)^m$

Solution: Ans: D the examples are i.i.d.,

$$\begin{aligned}
\mathbf{P}(\text{no mistakes on all } m \text{ examples}) &= \prod_{i=1}^{m} \mathbf{P}(\text{no mistake on } x_i) && (1)\\
&= \prod_{i=1}^{m} \mathbf{P}_{x\sim\mathcal{D}}\Big(h^*_{(t^*,r^*)}(x) \neq h_{(t_a,r_a)}(x)\Big) && (2)\\
&= (1-\epsilon)^m && (3)
\end{aligned}$$

(d) **(6 pts)** Show that the VC dimension $VC(\mathcal{H}_{ring}) \geq 2$ by completing the following proof.
Proof: To show $VC(\mathcal{H}_{ring}) \geq 2$, we consider the following two points:

S_1 :_____ and S_2 :_____.

$\mathcal{H}_{ring}$ can shatter these two points because for all the following label combinations, we can find the corresponding t, r such that $h_{t,r}(X)$ classifies S_1 and S_2, correctly.

S_1 Label	S_2 Label	t	r
+	+		
+	-		
-	+		
-	-		

Solution:

S_1: (1,0), S_2: (2,0)

S_1 Label	S_2 Label	t	r
+	+	1	2
+	-	1	1
-	+	2	2
-	-	0	0

(e) **(4 pts)** The VC dimension of $\mathcal{H}_{ring}$ in $\mathbb{R}^2$ is the same as which of the following problem(s) in $\mathbb{R}$? Select all the correct options.

(A) Positive ray classifier $\mathcal{H} = \{h_{(a)} | a \in \mathbb{R}\}$, where

$$h_{(a)}(x) = \begin{cases} 1, & \text{if } x \geq a \\ 0, & \text{otherwise} \end{cases}.$$

(B) Positive interval classifier $\mathcal{H} = \{h_{(a,b)} | a, b \in \mathbb{R}, a \leq b\}$, where

$$h_{(a,b)}(x) = \begin{cases} 1, & \text{if } a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}.$$

(C) Positive ray + positive interval classifier $\mathcal{H} = \{h_{(a,b,c)} | a, b, c \in \mathbb{R}, a \leq b \leq c\}$, where

$$h_{(a,b,c)}(x) = \begin{cases} 1, & \text{if } a \leq x \leq b \\ 1, & \text{if } x \geq c \\ 0, & \text{otherwise} \end{cases}.$$

(D) Double positive interval classifier $\mathcal{H} = \{h_{(a,b,c,d)} | a, b, c, d \in \mathbb{R}, a \leq b \leq c \leq d\}$, where

$$h_{(a,b,c,d)}(x) = \begin{cases} 1, & \text{if } a \leq x \leq b \\ 1, & \text{if } c \leq x \leq d \\ 0, & \text{otherwise} \end{cases}.$$

Solution: B

6 Support vector machines & kernel trick [15 pts]

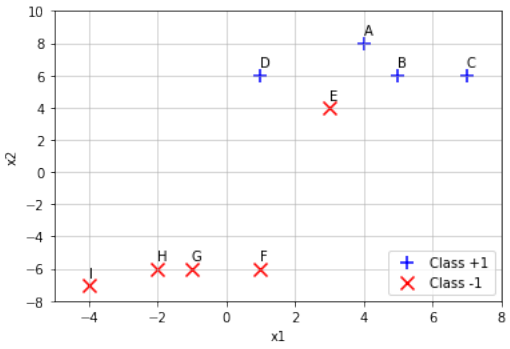
(a) **(4 pts)** Let $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^n$ be the input feature vectors. Let ϕ_a: $\mathbb{R}^n \to \mathbb{R}^k$ and ϕ_b: $\mathbb{R}^n \to \mathbb{R}^k$ be two feature transform functions. Consider two kernels $K_a(\boldsymbol{x}, \boldsymbol{y}) = \phi_a^T(\boldsymbol{x})\phi_a(\boldsymbol{y})$ and $K_b(x, y) = \phi_b^T(\boldsymbol{x})\phi_b(\boldsymbol{y})$. Let us define a new kernel $K_c(\boldsymbol{x}, \boldsymbol{y}) = 3K_b(\boldsymbol{x}, \boldsymbol{y}) + 4$ and let ϕ_c: $\mathbb{R}^n \to \mathbb{R}^{k+1}$ be its corresponding feature transformation. Write down the transformation ϕ_c in terms of ϕ_a and ϕ_b.

Solution: $\phi_c(\boldsymbol{x}) = [\sqrt{3}\phi_b(\boldsymbol{x}), 2]$.

(b) **(4 pts)** Continue 6(a). Now, consider $K_d(\boldsymbol{x}, \boldsymbol{y}) = K_a(\boldsymbol{x}, \boldsymbol{y})(K_b(\boldsymbol{x}, \boldsymbol{y}) + 1)$ and let ϕ_d: $\mathbb{R}^n \to \mathbb{R}^{k^2+k}$ be its corresponding feature transformation. Write down the transformation ϕ_d in terms of ϕ_a and ϕ_b.

Hint: you can use $\phi_a \times \phi_b$ to represent the Cartesian product between ϕ_a and ϕ_b. If $\phi_a = [1, 2, 3], \phi_b = [1, 2]$, $\phi_a \times \phi_b = [\phi_{a1}\phi_{b1}, \phi_{a2}\phi_{b1}, \phi_{a3}\phi_{b1}, \phi_{a1}\phi_{b2}, \phi_{a2}\phi_{b2}, \phi_{a3}\phi_{b2}] = [1, 2, 3, 2, 4, 6]$.

Solution: $\phi_c(\boldsymbol{x}) = [\phi_a(\boldsymbol{x}) \times \phi_b(\boldsymbol{x}), \phi_a(\boldsymbol{x})]$



Point	(x_1, x_2)	Label
A	(4,8)	+1
B	(5,6)	+1
C	(7,6)	+1
D	(1,6)	+1
E	(3,4)	-1
F	(1,-6)	-1
G	(-1,-6)	-1
H	(-2,-6)	-1
I	(-4,-7)	-1

Figure 1: Dataset

(c) **(4 pts)** Figure 1 provides a labelled training data in a 2-D plane. We learn a *hard SVM* classifier using this data. Answer the following questions.

Which of the following statement(s) are True? Select all of them.

☐ (A) There are exactly 4 support vectors - B, C, D & E

☐ (B) There are exactly 3 support vectors - A, B, & G

☐ (C) The decision boundary of the hard SVM classifier will be $x_2 = 0$

☐ (D) The decision boundary of the hard SVM classifier will be $x_2 = 5$

☐ (E) The training error of the hard SVM classifier is Zero.

☐ (F) There will be no feasible solution for hard margin SVM optimization problem if we flip the label of point B.

Solution: Correct options: A, D, E, F. This problem can be solved by plotting the points on 2-D plane and inspecting. Support Vectors: B, C, D, & E. Decision boundary $x_2 = 5$. Training error is 0.

(d) **(3 pts)** Fill in the blanks (use A, B,. . . , I to represent the data points):

If we remove one data point _____ in Figure 1, the decision boundary of the hard SVM will

change to be $x_2 =$_____ and the support vectors will be _____.

Solution: Remove point E. Support Vectors: B, C, D, F, G, & H. So the number of support vectors is 6. Decision boundary $y = 0$.

7 Perceptron [9 pts]

(a) **(3 pts)** Consider training a Perceptron model $y = w_1x_1 + w_2x_2 + b$ in the 2-dimensional feature space. If Perceptron makes a mistake on the data point (x_1, x_2) with label y where $x_1, x_2 \in \mathbb{R},\ y \in \{-1, 1\}$. Write down the update rule of w_1, w_2, and b. (Hint: In the lecture, we show a version of Perceptron update rule where we augment the weight vector w with the bias term b. Write down the update rule for each of the element of the vector.)

$w_1 \leftarrow$ _____ blank #1 _____.

$w_2 \leftarrow$ _____ blank #2 _____.

$b \leftarrow$ _____ blank #3 _____.

Solution:

$$w_1 \leftarrow w_1 + yx_1$$

$$w_2 \leftarrow w_2 + yx_2$$

$$b \leftarrow b + y$$

(b) **(6 pts)** Please fill the blanks to complete the proof of the following mistake bounds: Given a linear separable dataset $\mathcal{D} = \{(x_1^{(i)}, x_2^{(i)}), y^{(i)}\}$ $(-1 \le x_1^{(i)}, x_2^{(i)} \le 1,\ y^{(i)} \in \{-1, 1\}.)$ with margin γ, i.e., there exists a linear function $y = w_1^*x_1 + w_2^*x_2 + b^*$ satisfying

$$w_1^{*2} + w_2^{*2} + b^{*2} = 1,$$

$$\forall i,\ y^{(i)}(w_1^*x_1^{(i)} + w_2^*x_2^{(i)} + b^*) \ge \gamma.$$

If the Perceptron model is initialized as $w_1 = w_2 = b = 0$, prove that the Perceptron algorithm will make no more than $\frac{3}{\gamma^2}$ mistakes when training on D.

Proof: We denote $\theta = (w_1, w_2, b)$ as the weights, $\theta^{(k)}$ as the weights after making k mistakes, particularly, $\theta^{(0)} = \mathbf{0}$, and $\theta^* = (w_1^*, w_2^*, b^*)$.

Consider the inner product of $\theta^{(k)}$ and θ^*. Let j be the data point of the $(k+1)$-th mistake, and we have

$\theta^{(k+1)} \cdot \theta^* = \theta^{(k)} \cdot \theta^* +$ [_____ *blank#1* _____] $\ge \theta^{(k)} \cdot \theta^* +$ [_____ *blank#2* _____].

Since $\theta^{(0)} \cdot \theta^* = 0$,

$$\theta^{(k)} \cdot \theta^* \ge k\gamma.$$

Consider the l2-norm of $\theta^{(k)}$.

$$\|\theta^{(k+1)}\|^2 = \|\theta^{(k)}\|^2 + [\underline{\qquad blank\#3 \qquad}] \leq \|\theta^{(k)}\|^2 + 0 + 3.$$

Thus,

$$\|\theta^{(k)}\|^2 \leq 3k,$$
$$k\gamma \leq \theta^{(k)} \cdot \theta^* \leq \|\theta^{(k)}\|\|\theta^*\| \leq \sqrt{3k},$$
$$k \leq \frac{3}{\gamma^2}.$$

Solution:
Blank1: $y^{(j)}(w_1^* x_1^{(j)} + w_2^* x_2^{(j)} + b^*)$, or $\theta^* \cdot y^{(j)}\mathbf{x}^{(j)}$. Here $\mathbf{x}^{(j)} = (x_1^{(j)}, x_2^{(j)}, 1)$
Blank2: γ. Some equivalent replacements as Blank1 are not regarded as correct answers.
Blank3: $2y^{(j)}(w_1^{(k)} x_1^{(j)} + w_2^{(k)} x_2^{(j)} + b^{(k)}) + (x_1^{(j)2} + x_2^{(j)2} + 1)$, or $2\theta^{(k)} \cdot y^{(j)}\mathbf{x}^{(j)} + \|\mathbf{x}^{(j)}\|^2$. It's also correct to add a coefficient $y^{(j)2}$ for the second term because it is 1.