CSCI 567 Fall 2022 Practice Problems for Quiz 1

Instructor: Vatsal Sharan

1 Multiple choice questions (30 points)

IMPORTANT: *Select ALL options among {A,B,C,D} that you think are correct. You get 0.5 point for selecting each correct option and similarly 0.5 point for not selecting each incorrect option. You get 1 additional point for selecting all four options correctly.*

(1) Which of the following are true statements about machine learning?

(A) Cross-validation is often used to tune the hyper-parameters of a machine learning algorithm.
(B) The goal of a machine learning algorithm is to achieve zero error on a training set.
(C) Regularization is a common way to prevent overfitting.
(D) Classification and regression are two common tasks in machine learning.

Answer: ACD

(2) Which of the following phenomenon is called overfitting?

(A) low training error, low test error
(B) high training error, high test error
(C) high training error, low test error
(D) low training error, high test error

Answer: D

A: $(X^TX+\lambda I)^{-1}X^Ty$

B: $X=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \end{bmatrix}_{n\times d}$ $\quad n = 3,\ d = 3$

$\mathrm{rank}(X)=2$

$\mathrm{rank}(X^TX)\le 2$

(3) Which of the following are true statements about linear regression?

(A) The least square solution has a closed-form formula, even if ℓ_2 regularization is applied.
(B) The covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible if and only if the number of data points n is smaller than the dimension d. $\Leftarrow$ ✓ $\Rightarrow$ ×
(C) When the covariance matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is not invertible, the Residual Sum of Squares (RSS) objective has no minimizers.
(D) Linear regression with kernels is a non-parametric method.

C: multiple

Answer: AD

(4) Consider a training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ and a probabilistic model $\mathbb{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{w})$ which specifies for each i the probability of seeing outcome y_i given feature $\boldsymbol{x}_i$ and parameter $\boldsymbol{w}$. Which of the following is the Maximum Likelihood Estimation (MLE) for $\boldsymbol{w}$?

(A) $\mathrm{argmax}_{\boldsymbol{w}} \sum_{i=1}^n \mathbb{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{w})$ (B) $\mathrm{argmax}_{\boldsymbol{w}} \Pi_{i=1}^n \mathbb{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{w})$
(C) $\mathrm{argmax}_{\boldsymbol{w}} \sum_{i=1}^n \ln \mathbb{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{w})$ (D) $\mathrm{argmax}_{\boldsymbol{w}} \Pi_{i=1}^n \ln \mathbb{P}(y_i \mid \boldsymbol{x}_i; \boldsymbol{w})$

Answer: BC

(5) Which of the following are true statements about kernels?

A: $\phi(x)\cdot\phi(x') = k(x,x')$

(A) A machine learning algorithm can be kernelized if it does not need explicit access to the feature vectors and instead only requires access to inner products of the feature vectors.
(B) A Gram/kernel matrix must be positive semidefinite.

Verify kernel

(D) ① PSD $K \Leftrightarrow u^T K u \geq 0$

$k(\cdot,\cdot) = k_1(\cdot,\cdot) + k_2(\cdot,\cdot)$

$k = k_1 + k_2$

$u^T k u = u^T k_1 u + u^T k_2 u \geq 0$

(C) ② construct $\phi(x)^T \cdot \phi(x')$

$\phi(x)$ $= K(x, x')$

$\Rightarrow$ $= k_1(x,x') \cdot k_2(x,x')$

$= \phi_1^T(x)\, \phi_1(x')$

$\phi_2^T(x)\, \phi_2(x')$

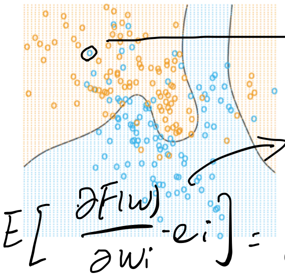
$= \cdots$ expand

(C) The product of two kernel functions is still a kernel function.
(D) The sum of two kernel functions is always a kernel function.

C: $E[-\nabla f_i(w_0) + \nabla F(w_0)] = 0$

Answer: ABCD

(6) Which of the following ML models could have generated the decision boundary in the binary classification problem on the right?
(A) SVM without kernels
(B) 1-nearest-neighbor.
(C) Logistic regression without kernels
(D) None of the above.



$\boxed{\sum_i}$

$\begin{bmatrix} 0 \\ \vdots \\ \partial F(w)/\partial w_i \\ \vdots \\ 0 \end{bmatrix}_{d\times 1}$

Answer: D

$D: E\left[\boxed{\frac{\partial F(w)}{\partial w_i}} \cdot d e_i\right] = d\, E\left[\frac{\partial F(w)}{\partial w_i} \cdot e_i\right] = d \cdot \frac{1}{d} \cdot \sum_i (\ \ )$

(7) A machine learning objective function is usually of the form $F(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{w})$ for some loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ corresponding to the i-th training point. We know that $\nabla f_i(\boldsymbol{w})$ for i chosen uniformly at random is a stochastic gradient of this objective since $\mathbb{E}[\nabla f_i(\boldsymbol{w})] = \nabla F(\boldsymbol{w})$. Which of the following is also a stochastic gradient?

$= \begin{bmatrix} \partial F(w)/\partial w_1 \\ \vdots \\ \partial F(w)/\partial w_d \end{bmatrix} = \nabla F(w)$

B:

$E\left[\frac{1}{|S|}\sum_{i\in S} \nabla f_i(w)\right]$

(A) $\sum_{i\in S} \nabla f_i(\boldsymbol{w})$ for a subset $S \subset \{1,\ldots,n\}$ (of some fixed size) chosen uniformly at random.
(B) $\frac{1}{|S|}\sum_{i\in S} \nabla f_i(\boldsymbol{w})$ for a subset $S \subset \{1,\ldots,n\}$ (of some fixed size) chosen uniformly at random.
(C) $\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}_0) + \nabla F(\boldsymbol{w}_0)$ for i chosen uniformly at random and some fixed point $\boldsymbol{w}_0$.
(D) $\frac{\partial F(\boldsymbol{w})}{\partial w_i} \cdot d\boldsymbol{e}_i$ for a coordinate i chosen uniformly at random ($\boldsymbol{e}_i \in \mathbb{R}^d$ is the standard basis vector with 1 in the i-th coordinate and 0 in all other coordinates).

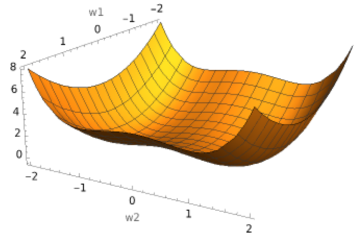
$\frac{1}{|S|} E\left[\sum_{i\in S} \nabla f_i(w)\right]$

$\frac{1}{|S|} \cdot |S| \cdot E[\nabla f_i(w)]$

$|S| \nabla F(w)$

Answer: BCD

(8) Consider a two-dimensional function $F(\boldsymbol{w}) = w_1^2 - w_2^2 + \frac{1}{2}w_2^4$ (a plot is provided below). Which of the following statement is correct?



(A) F is convex.
(B) F has three stationary points: $(0,0)$, $(0,-1)$, and $(0,1)$.
(C) $(0,0)$ is a saddle point of F.
(D) Gradient Descent always converges to $(0,-1)$, regardless of initialization.

$2w_1 = \frac{\partial F(w)}{\partial w_1} = 0$

$= \frac{\partial F(w)}{\partial w_2}$

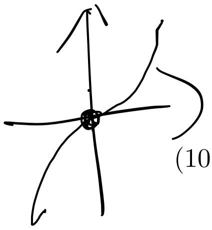
$2w_2^3 - 2w_2 = 0$

Answer: BC

$y \overset{?}{=} \mathrm{sgn}(w^T x)$

$(x, y) \sim D_{test}$

(9) Which of the following is true?
(A) Normalizing the output $\boldsymbol{w}$ of the perceptron algorithm so that $\|\boldsymbol{w}\|_2 = 1$ changes its test error.
(B) Normalizing the output $\boldsymbol{w}$ of the perceptron algorithm so that $\|\boldsymbol{w}\|_1 = 1$ changes its test error.
(C) Minimizing 0-1 loss is NP-hard for every dataset, even if it is linearly separable.
(D) Perceptron algorithm may fail to converge on linearly-separable data.



Answer: C

(10) Which of the following is not a kernel function?

(A) $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}')^2$

(B) $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + 1)^2$) polynomial

(C) $k(\boldsymbol{x}, \boldsymbol{x}') = -\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2^2$

(D) $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2^2)$ — RBF

Answer: C

2 Short answer questions

2.1 Linear Classifiers with squared hinge loss (6 points)

In class, we have seen the hinge loss $\ell(z) = \max\{0, 1 - z\}$. In this question, we will consider a modification of the hinge loss. Given a binary dataset $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \{-1, 1\}$, we define the following new loss function for a linear model $\boldsymbol{w} \in \mathbb{R}^d$:

$$F(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \quad \text{where } f_i(\boldsymbol{w}) = \left(\max\left\{0, 1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\right\}\right)^2. \tag{1}$$

For a fixed i, write down the gradient $\nabla f_i(\boldsymbol{w})$ (show your derivation), then fill in the missing details in the repeat-loop of the algorithm below which applies SGD to minimize F.

Algorithm 1: SGD for minimizing Eq. (1)

```
Input: A training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \{-1, 1\}$, learning rate $\eta > 0$
Initialization: $\boldsymbol{w} = \boldsymbol{0}$
Repeat:
    randomly pick an example $(\boldsymbol{x}_i, y_i)$
    compute $z = y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i$ and update $\boldsymbol{w} \leftarrow \boldsymbol{w} + 2\eta y_i \max\{0, 1 - z\}\boldsymbol{x}_i$
```

When $1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i < 0$, the function is a constant and thus the gradient is $\boldsymbol{0}$. On the other hand, when $1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i > 0$, the function is simply $f_i(\boldsymbol{w}) = (1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)^2$, and thus by chain rule we have $\nabla f_i(\boldsymbol{w}) = -2y_i(1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)\boldsymbol{x}_i$, which is also $\boldsymbol{0}$ when $y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i$ approaches 1. Therefore, we have

$$\nabla f_i(\boldsymbol{w}) = -2y_i \max\{0, 1 - y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\}\boldsymbol{x}_i.$$

Rubrics:

- 3 points for the derivation of the gradient. Technically we need some discussion for the case $y_i \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i = 1$ as the solution above shows, but for simplicity it is okay if such discussion is missing as long as the final gradient is correct.
- 2 points for the SGD implementation. Do not deduct more points for wrong gradient from the earlier wrong derivation.

2.2 Nearest Neighbor Classification (4 points)

In Homework 1, we used the Euclidean distance (ℓ_2 distance) and the cosine distance to find the nearest neighbors for the k-NN algorithm. In this question we will relate the two notions. Let us recall the definitions of Euclidean distance and cosine distance. The Euclidean distance $E(\mathbf{x}, \mathbf{x}')$ is defined as

$$E(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||_2 = \sqrt{\sum_{j=1}^{d} (x_j - x_j')^2}. \tag{2}$$

The cosine distance $C(\mathbf{x}, \mathbf{x}')$ is defined as (assuming $\|\boldsymbol{x}\| \neq 0$ and $\|\boldsymbol{x}'\| \neq 0$),

$$C(\mathbf{x}, \mathbf{x}') = 1 - \frac{\mathbf{x}^T \mathbf{x}'}{||\mathbf{x}||_2 ||\mathbf{x}'||_2} = 1 - \frac{\sum_{j=1}^{d} (x_j \cdot x_j')}{||\mathbf{x}||_2 ||\mathbf{x}'||_2}. \tag{3}$$

Show that if all the datapoints are normalized to have unit ℓ_2 norm (that is, $\|\boldsymbol{x}\| = 1$ for all $\mathbf{x}$ in the training and test sets) then changing the distance function from the Euclidean distance to the cosine distance will NOT affect the nearest neighbor classification results.

The goal is to construct a function $C(\mathbf{x}, \mathbf{x}') = f(E(\mathbf{x}, \mathbf{x}'))$ where f is monotonic when $E(\mathbf{x}, \mathbf{x}') \geq 0$. Once we have a monotonic f, for any point $\mathbf{x}$, $E(\mathbf{x}, \mathbf{x_1}) \geq E(\mathbf{x}, \mathbf{x_2}) \iff C(\mathbf{x}, \mathbf{x_1}) \geq C(\mathbf{x}, \mathbf{x_2})$, k-NN will retrieve the same neighbors with two distance functions.

Given $\|\boldsymbol{x}\| = 1$ for all $\mathbf{x}$, we have $C(\mathbf{x}, \mathbf{x}') = 1 - \frac{\mathbf{x}^T \mathbf{x}'}{||\mathbf{x}||_2 ||\mathbf{x}'||_2} = 1 - \mathbf{x}^T \mathbf{x}'$. Try to construct a inner product term from $E(\mathbf{x}, \mathbf{x}')$. $E(\mathbf{x}, \mathbf{x}')^2 = ||\mathbf{x}||_2^2 + ||\mathbf{x}'||_2^2 - 2\mathbf{x}^T \mathbf{x}' = 2 - 2\mathbf{x}^T \mathbf{x}' = 2C(\mathbf{x}, \mathbf{x}')$.

Rubrics:

- 3 points for constructing the relationship between two distance functions.
- 1 point for the monotonicity explanation, other reasonable explanation also works.

2.3 ML diagnostics: Gradient descent (5 points)

Let $F(\boldsymbol{w}) = w_1^2 + 2w_2^2$.

(1) Write down the gradient descent update rule for minimizing $F(\boldsymbol{w})$.

$$\frac{\partial F(\boldsymbol{w})}{\partial \boldsymbol{w}} = \begin{bmatrix} \partial F(\boldsymbol{w})/\partial w_1 \\ \partial F(\boldsymbol{w})/\partial w_2 \end{bmatrix} = \begin{bmatrix} 2w_1 \\ 4w_2 \end{bmatrix}$$

$$\boldsymbol{w} = \boldsymbol{w} - \eta \frac{\partial F(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \eta \begin{bmatrix} 2w_1 \\ 4w_2 \end{bmatrix}, \text{where } \eta \text{ is the learning rate.}$$

The figure below depicts a run of gradient descent to minimize $F(\boldsymbol{w})$ (initializing at $w_1 = w_2 = 40$, and running for 20 iterations). The value of $\boldsymbol{w}$ after every update is shown as a red circle. The contour lines of $F(\boldsymbol{w})$ are also depicted in the figure.

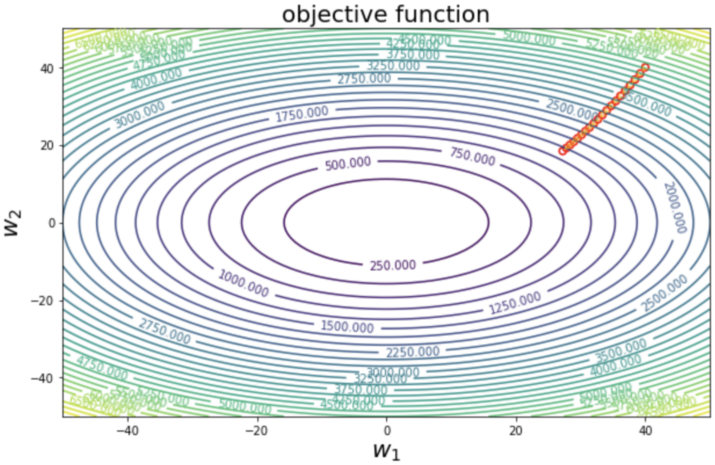


Figure 1: Gradient descent on $F(\boldsymbol{w}) = w_1^2 + 2w_2^2$.

(2a) If we continued to run gradient descent for a sufficiently large number of iterations, to what point would it converge to?

(0, 0)

(2b) Does it seem possible to make gradient descent converge faster than in the figure (with the same initialization and for the function $F(\boldsymbol{w})$)? Explain.

Increase learning rate.

Rubrics:

- 3 points for (1). Correct gradient of $F(\boldsymbol{w})$ (2 points), the update function (1 point). Writing two update functions for w_1 and w_2 separately is also fine.

- 1 point for (2a).
- 1 point for (2b).

3 Linear Regression (8 points)

In class, we discussed that if we use Newton's method to solve the least square optimization problem then it only takes one step to converge. We will prove this statement in this problem. Let

$$F(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2$$

where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Recall that Newton's method updates the parameters as follow:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)})$$

where $\boldsymbol{H}_t = \nabla^2 F(\boldsymbol{w}^{(t)}) \in \mathbb{R}^{d\times d}$ is the Hessian matrix of the objective function evaluated at $\boldsymbol{w}^{(t)}$, i.e. for every index $u, v \in \{1, \ldots, d\}$ the (u, v)-th entry of $\boldsymbol{H}_t$ is $H_t(u, v) = \frac{\partial^2}{\partial w_u \partial w_v}F(\boldsymbol{w}) \mid_{\boldsymbol{w}=\boldsymbol{w}^{(t)}}$.

(1) Find the Hessian of the least square objective function $F(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2$.

$$\begin{aligned}\frac{\partial}{\partial w_u}F(\boldsymbol{w}) &= \sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)x_{iu}\\ \frac{\partial^2}{\partial w_u \partial w_v}F(\boldsymbol{w}) &= \sum_{i=1}^{n}x_{iu}x_{iv} = (\boldsymbol{X}^T\boldsymbol{X})_{uv}\end{aligned}$$

Therefore, $\boldsymbol{H} = \nabla^2 F(\boldsymbol{w}) = \boldsymbol{X}^T\boldsymbol{X}$.

(2) Show that given any initialization $\boldsymbol{w}^{(0)}$, after one iteration of Newton's method we obtain the optimal $\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$. (Here $\boldsymbol{X}$ is the $n \times d$ data matrix with one row per data point, and $\boldsymbol{y}$ is the n-vector of their labels.)

$$\begin{aligned}\boldsymbol{w}^{(1)} &= \boldsymbol{w}^{(0)} - \boldsymbol{H}^{-1}\nabla L(\boldsymbol{w}^{(0)})\\ &= \boldsymbol{w}^{(0)} - \boldsymbol{H}^{-1}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}^{(0)} - \boldsymbol{y})\\ &= \boldsymbol{w}^{(0)} - \boldsymbol{w}^{(0)} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\end{aligned}$$

Rubrics:

- 4 points for (1).
- 4 points for (2).