Name **Student ID**

1 Question-answering (71)

Provide short answers to the questions below.

1. (1) What is the difference between the population loss and empirical loss?
   Answer:
   Population loss is the loss on the real distribution.
   Empirical loss is the loss on the training set.

2. (1) In research papers, authors only display the results for training loss/test loss, without population loss. Why?
   Answer:
   Because we can only estimate the population loss, we can not get the exactly real distribution/we can not compute it directly.

3. (1) What is the difference between regression and classification?
   Answer:
   Regression's label is continuous and Classification's lable is discrete.

4. (2) ReCAPTCHA displays two words for the users to type. Why?
   Answer:
   One have label, and the other do not have label. If you answer the first correctly, you have a high probability to answer the other correctly. And then we can label the second word.

5. (2) How do you use a generative model to describe a bizarre distribution, given samples from this distribution?
   Answer:
   A generative model maps a Gaussian to that distribution. By the mapping, we describe the bizarre distribution.

6. (2) What is the smoothness assumption?
   Answer:
   L-Smoothness: $|f(w) - f(w^{'}) - < \nabla f(w^{'}), w - w^{'} > | \leq L/2|w - w^{'}|$

7. (2) Stochastic gradients are always computed using mini-batch, is it true? Why is that?
   Answer:
   False, we can also use full gradients and noise(zero mean) to compute Stochastic gradients. It is also an unbias estimation.

8. (2) What is measurement matrix A for compressed sensing? Give one example.
   Answer:
   $y = Ax$, we know x is sparse. A map x to a low dimension, and we can use y and A to recover x. We can choose A.
   One example: Compressed sensing image reconstruction.(Any Reasonablr example.)

9. (2) What is the main difference between the settings of compressed sensing and Lasso?
   Answer:
   In compressed sensing, we know x is sparse, we choose A and use y=Ax to recover x.

In Lasso, we have a fixed A(train set input), y is the label. We want to find a sparse w that y = Aw.

10. (1) In the dual program of SVM, we are optimizing the dual variable a. What is the dimension of a?
Answer:
number of the train data.

11. (2) When the cross entropy loss is minimized, what is its value?
Answer:
The Entropy of label.

12. (2) What is power method?
Answer:
Power iteration can solve the eigenvalue with the largest absolute value.

$$
\begin{aligned}
&\text{init } b_0 \\
&\text{for } i = 1, 2... \\
&\quad u_k = Ab_{k-1} \\
&\quad b_k = u_k / ||u_k|| \\
&\text{end}
\end{aligned}
$$

13. (2) Why do we need back-pressure in streaming programing?
Answer:
In the pipeline, if the processing speed of the upper stage is too fast, the next stage will accumulate a lot of unfinished tasks, and the next stage will feed back to the upper stage through a back pressure mechanism to more efficiently adjust the overall jobs.

14. (3) Why do we say k-NN is a non-parametric algorithm? Empirically, if we pick a small k, what might be the problem? If we pick a large k, what might be the problem?
Answer:
There is only one hyperparameter in k-NN, and there are no any other parameters need to be trained.
Small k: Overfit, non-smooth.
Large k: May consider too much data not related to this point.(can not capture the pattern clearly; When k=N, the class with more data wins.)

15. (2) What does (R, cR, P_1, P_2) sensitive mean for LSH family?
Answer:
For points p and q:
if $||p - q|| \leq R$, then $p[h(p) = h(q)] \geq P_1$
if $||p - q|| \geq cR$, then $p[h(p) = h(q)] \leq P_2$

16. (2) Give one LSH familly under Hamming distance for $\{0, 1\}^d$, and compute the corresponding (R, cR, P_1, P_2) parameters for given R and cR.
Answer:
$P_1 = 1 - \frac{R}{d}, P_2 = 1 - \frac{cR}{d}$

17. (2) What is NCA algorithm? Why do we say it is a relaxtion of k-NN?
Answer:
for point x_i and x_j, define $p_{i,j} = \frac{e^{-||f(x_i)-f(x_j)||}}{\sum_{i\neq j} e^{-||f(x_i)-f(x_j)||}}$
Then NCA optimizes $\sum_i \sum_{j\neq i, j\in C_i} p_{i,j}$
It is a relation of k-NN(soft-version).

18. (3) In Professor Huang's guest lecture, he mentioned a technique to train a neural network of depth 1202 layers. Describe this technique.
Answer:
Stochastic Depth. Random drop some block layers.

19. (2) If we write a boolean function $f : \{-1, 1\}^n \to [0, 1]$ in the Fourier basis, it is:

$$f(x) = \sum_{S\subseteq[n]} \hat{f}_S \chi_s(x)$$

How to compute $\hat{f_S}$?
Answer:
$\hat{f_S} = E_{x\sim D} f(x)\chi_s(x)$

20. (3) What is KM algorithm in the Boolean analysis? (You do not need to write down how to estimate $E[f_\alpha^2]$)
Answer:
KM(α):
if $[f_\alpha^2 \leq \epsilon]$ return
if α is of length n: KM(α_0), KM(α_1)
else return

21. (3) When proving the generalization guarantees of Adaboost, the authors modified the function $f(x)$ from $\sum_{t=1}^T a_t h_t(x)$ to $\frac{\sum_{t=1}^T a_t h_t(x)}{\sum_{t=1}^T a_t}$. Does that change the Adaboost algorithm's prediction? What has been changed after this modification on $f(x)$?
Answer:
No, Because the prediction is based on the sign.
It is helpful for the analysis of generalization. Margin is changed.

22. (2) What is Danskin's theorem (only the expression is needed)?
Answer:
$\frac{\partial}{\partial\theta} \max_{\delta\in\Delta} L(f(x+\delta), y) = \frac{\partial}{\partial\theta} L(f(x+\delta^*), y)$ Where $\delta^* = \arg\max_{\delta\in\Delta} L(f(x+\delta, y)$

23. (2) How to attack a model, if the gradient of the model cannot be explicitly computed?
Answer:
Say the model is $f(g(x))$, g is not differentiable, since usually $g(x) \approx x$, we just use $\nabla f(x)$ to approximate.

24. (2) In the randomized smoothing defense, if we use a uniform distribution over the ball $B_r(0)$ as the smoothing kernel, given the base classifier f (for simplicity, assume $f(\cdot) \in \mathbb{R}$), please write down the prediction of the smoothed classifier g (assume $g(\cdot) \in \mathbb{R}$) at x and $x + \delta$.
Answer:

$g(x) = \int\limits_{v \in Br(0)} \frac{1}{VolBr(0)} f(x+v)dv$
$g(x+\delta) = \int\limits_{v \in Br(0)} \frac{1}{VolBr(0)} f(x+v+\delta)dv$

25. (2) In the randomized smoothing defense, if you not only get the histogram of $g(x)$, but also the histogram of $g(x')$ where x' is a close neighbor of x. Theoretically, would that change your robustness guarantees? Why? Will the guarantee become better/worse/unclear?
Answer:
Yes, Maybe Better. Because robust guarantee is the worst case near x. In fact, the guarantee may be better because we can further make robust guarantee around $x^{'}$.

26. (2) Describe the adversarial training algorithm.
Answer:
Sample mini-batch data
Generate Adv examples use PGD or FGSM
Use Adv examples to train the model
repeat the steps

27. (2) What are the two limitations of Bayesian optimization?
Answer:
1, Can not parallel.
2, Need Prior Distribution
3, Can not handle High Dimension case

28. (2) In the analysis of SH algorithm for hyperparamter tuning, we defined $\bar{\gamma}$ and γ_i. What are the exact definitions?
Answer:
Let $\gamma_i(t)$ be non-increasing function of t, which gives the smallest value for each t $s.t. ||\ell_{i,j} - \nu_i|| \leq \gamma_i(t)$
Let $\gamma_i^{-1}(\alpha) = min\{t \in N : \gamma_i(t) \leq \alpha\}$
$\hat{\gamma} = \max\limits_i \gamma_i(t)$

29. (2) In ProxylessNAS algorithm, we have parameters $\{\alpha_i\}$ for component i in the current layer. Why can't we just compute $\partial L/\partial \alpha_i$? How did we solve the problem?
Answer:
It is not fully diffierentable
use $\frac{\partial L}{\partial \alpha_i} \approx \sum\limits_j \frac{\partial L}{\partial g_j} \frac{\partial p_j}{\partial \alpha_i}$

30. (2) What is the non-convex formulation of matrix completion problem? Why is it non-convex?
Answer:
minimize $||\Omega(UV) - \Omega(A)||_F^2$
Because $(U^*, V^*) and (\frac{U^*}{k}, kU^*)$ are equla minima.

31. (2) What is the graph Laplacian of a circle of 4 nodes with unit weights (assume the circle is undirected)?
Answer:
$$\begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}$$

32. (3) Assume L is the graph Laplacian of an undirected graph, w_{ij} is the weight of edge between node i and node j, v is any vector in $\mathbb{R}^n$. Prove $v^T L v = \frac{1}{2}\sum_{ij} w_{ij}(v_i - v_j)^2$.
Answer:
$v^T L v = v^T D v - v^T A v$

$$= \sum_{i=1}^{n} d_i v_i^2 - \sum_{i,j}^{n} v_i v_j w_{i,j}$$

$$= \frac{1}{2}(\sum_{i=1}^{n} d_i v_i^2 - 2\sum_{i,j} v_i v_j w_{i,j} + \sum_{j=1}^{n} d_j v_j^2)$$

$$= \frac{1}{2}\sum_{ij} w_{ij}(v_i - v_j)^2$$

33. (2) Why do we use exponential search instead of binary search for learned indexed structure?
Answer:
Use Binary Search waste too much time Because the prediction is close to the ground truth.

34. (2) What is the formal definition of differential privacy?
Answer:
A randomized algorthm M with domain $N^{|X|}$ is (ϵ, δ)-differetially private if for all $S \subseteq Range(M)$ and for all $x, y \in N^{|X|}$ such that $||x - y||_1 \leq 1$:
$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta$

35. (2) When computing decision trees using gini index, for a given feature (assuming it has two possible values Yes/No), how to compute its gini index?
Answer:
$Gini = P(Yes)(1 - \sum P(y_i|Yes)^2) + P(No)(1 - \sum P(y_i|No)^2)$

2 SNE algorithm (3)

In SNE algorithm, the authors used KL divergence to measure the differences between the original and target distributions.

- What is the definition of KL divergence?
- KL divergence is asymmetric, so it may emphaisize one kind of error and ignore another kind of error during the mapping. Please describe these two kinds of errors.
- How did t-SNE solve this problem?

Answer 1.

$$L = \sum_i KL\left(P_i||Q_i\right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

(Or vanilla KL divergence.)

2. If $p_{j|i}$ is large, $q_{j|i}$ is small, the loss will be emphasized. If $p_{j|i}$ is small, $q_{j|i}$ is small large, the loss will be ignored.

3. Use student-T distribution. (Or other reasonable answers.)

3 Neural network(4)

In the analysis two layer neural network, please write down the update rule of $f_{W(t)}(X) - y$ using matrix $H(t) = Z(t)^T Z(t)$. What can't we directly get convergence result using this updating rule? How to solve this problem?

Answer 1. $f_{W(t+1)}(X) - y = (I - \eta H(t))(f_{W(t)}(X) - y)$

2. Because H is changing over t.

3. We assume H is not changing if we are using a super wide neural network. (Or other reasonable answers.)

4 Spectral graph clustering (3)

What is ratio cut problem? For $k = 2$, why is spectral graph clustering a relaxation for ratio cut problem?

Answer Define $v^A = (v_1, v_2, \cdots, v_n) \in R^n$, s.t.

$$v_i^A = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}}, & i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}}, & \text{o. } w \end{cases}$$

We have

$$\begin{aligned} &\left(v^A\right)^T L v^A = \tfrac{1}{2} \textstyle\sum_{ij} w_{ij} \left(v_i^A - v_j^A\right)^2 \\ &= \tfrac{1}{2} \textstyle\sum_{i \in \mathrm{A}, j \in \bar{A}} w_{ij} \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 + \tfrac{1}{2} \textstyle\sum_{j \in \mathrm{A}, i \in \bar{A}} w_{ij} \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 \\ &= \mathrm{cut}(A, \bar{A}) \left(\tfrac{|\bar{A}|}{|A|} + \tfrac{|A|}{|\bar{A}|} + 2\right) = \mathrm{cut}(A, \bar{A}) \left(\tfrac{|\bar{A}| + |A|}{|A|} + \tfrac{|A| + |\bar{A}|}{|\bar{A}|}\right) \\ &\text{Notice that } |A| + |\bar{A}| = |V| \\ &= |V| \, \mathrm{ratiocut}(A, \bar{A}) \end{aligned}$$

Therefore

$$\min_A ratiocut(A, \bar{A} = \min_{A \subset V} (v^A)^T L v^A)$$

Notice that $v \perp I_V$, so the solution is the second smallest eigenvalue

5 Promise for user utilities(4)

Using the framework described in class, explain why differential privacy mechanism can promise that the utility of a given user will not change much, no matter what he/she reports in the mechanism.

Answer i has some preferences over set of future events, denoted as A. Assume $f : \mathrm{Range}(M) \to \Delta(A)$ be any function that determines the distribution of future events, based on M,

$$\begin{aligned} &E_{a \sim f(M(x))} \left[u_i(a)\right] = \textstyle\sum_{a \in A} u_i(a) \cdot \Pr_{f(M(x))}[a] \\ &\leq \textstyle\sum_{a \in \mathrm{A}} u_i(a) \cdot \exp(\epsilon) \Pr_{f(M(y))}[a] = \exp(\epsilon) E_{a \sim f(M(y))} \left[u_i(a)\right] \end{aligned}$$

Similarly,

$$E_{a \sim f(M(x))} \left[u_i(a)\right] \geq \exp(-\epsilon) E_{a \sim f(M(y))} \left[u_i(a)\right]$$

If M is DP, we know that i's utility will not be harmed by more than $exp(\epsilon) \sim (1 + \epsilon)$ factor

6 Adaboost (4)

Recall the boosting framework is defined as:

1. Given training set $(x_1, y_1), \cdots, (x_m, y_m)$, $y_i \in \{-1, +1\}, x_i \in X$.
2. For $t = 1, \cdots, T$:
   - Build distribution D_t on training data points $\{1, \cdots, m\}$.
   - Find weak learner $h_t : X \to \{-1, +1\}$ with error $\epsilon_t \triangleq \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
3. Output the final classifier H.

Specifically, Adaboost algorithm makes the following choices:

- $D_1(i) \triangleq \frac{1}{m}$ (uniform)
- Given D_t and h_t, $D_{t+1}(i) \triangleq \frac{D_t(i)}{Z_t} \cdot e^{-\alpha_t y_i h_t(x_i)}$
- $Z_t \triangleq \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h_t(x)}$
- $\alpha_t \triangleq \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$
- $f(x) \triangleq \sum_t \alpha_t h_t(x)$
- $H(x) = \text{sign}(f(x))$

Please prove

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(H(x_i) \neq y_i) \leq \Pi_{t=1}^{T} Z_t$$

Answer $G(x_i) \neq y(x_i) \to y(x_i) f(x_i) < 0 \to \exp(y(x_i) f(x_i)) < 1 \to \exp(-y(x_i) f(x_i)) > 1 \geq \mathrm{I}(G(x_i) \neq f(x_i)$

$$\begin{aligned}
\frac{1}{N} \sum_{i=1}^{N} \exp(-y_i f(x_i)) &= \frac{1}{N} \sum_{i=1}^{N} \exp\left(-y_i \sum_{m=1}^{M} \alpha_m G_m(x_i)\right) \\
&= \frac{1}{N} \sum_{i=1}^{N} \exp\left(\sum_{m=1}^{M} y_i \alpha_m G_m(x_i)\right) \\
&= \frac{1}{N} \sum_{i=1}^{N} \prod_{m=1}^{M} \exp(y_i \alpha_m G_m(x_i))
\end{aligned}$$

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\exp\left(-y_i f\left(x_i\right)\right) &= \frac{1}{N}\sum_{i=1}^{N} w_{1i}\prod_{m=1}^{M}\exp\left(y_i\alpha_m G_m\left(x_i\right)\right) \\
&= Z_1\frac{1}{N}\sum_{i=1}^{N} w_{2i}\prod_{m=2}^{M}\exp\left(y_i\alpha_m G_m\left(x_i\right)\right) \\
&= \cdots \\
&= \frac{1}{N}Z_1 Z_2\cdots Z_M\sum_{i=1}^{m} 1 \\
&= \frac{1}{N}Z_1 Z_2\cdots Z_M N \\
&= \prod_{m=1}^{M} Z_m
\end{aligned}$$

7 Rademacher complexity of linear classes (3)

Let $S = (x_1, \cdots, x_m)$ be vectors in a Hilbert space. Define $H_2 \circ S = \{(\langle w, x_1\rangle, \cdots, \langle w, x_m\rangle) : \|w\|_2 \leq 1\}$. Prove:

$$R(H_2 \circ S) \leq \frac{c\max_i \|x_i\|_2}{\sqrt{m}}$$

for some numeric constant c.

Answer

See Proof of Lemma 26.10 in Understanding Machine Learning, from Theory to Algorithms.

8 Finite hypothesis class is learnable (4)

Let H be a finite hypothesis class. Let $\delta \in (0,1)$ and $\epsilon > 0$, and let m be an integer that satisfies

$$m \geq \frac{\log(|H|/\delta)}{\epsilon}$$

Prove: for any labeling function f and for any distribution D, for which the realizability assumption holds, with probability of at least $1-\delta$ over the choice of an iid sample S of size m, we have that for every ERM hypothesis h_S, it holds that $L_{(D,f)}(h_S) \leq c\epsilon$ for some numeric constant c.

Answer See Proof of Corollary 2.3 in Understanding Machine Learning, from Theory to Algorithms.

9 SVRG(4)

Recall SVRG in Figure 1. Assume the function is L-smooth and μ strongly convex. If we already know that inside the inner loop, for every iteration t,

$$E\|w_t - w^*\|_2^2 \leq \|w_{t-1} - w^*\|_2^2 - 2\eta(1-2L\eta)(f(w_{t-1}) - f(w^*)) + 4L\eta^2(f(\tilde{w}) - f(w^*))$$

Please prove:

$$E[f(\tilde{w}_S) - f(w^*)] \leq \left[\frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta}\right] E[f(\tilde{w}_{s-1}) - f(w^*)]$$

Procedure SVRG

Parameters update frequency m and learning rate η
Initialize $\tilde{w}_0$
Iterate: for $s = 1, 2, \ldots$
  $\tilde{w} = \tilde{w}_{s-1}$
  $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \nabla\psi_i(\tilde{w})$
  $w_0 = \tilde{w}$
  Iterate: for $t = 1, 2, \ldots, m$
    Randomly pick $i_t \in \{1, \ldots, n\}$ and update weight
      $w_t = w_{t-1} - \eta(\nabla\psi_{i_t}(w_{t-1}) - \nabla\psi_{i_t}(\tilde{w}) + \tilde{\mu})$
  end
  option I: set $\tilde{w}_s = w_m$
  option II: set $\tilde{w}_s = w_t$ for randomly chosen $t \in \{0, \ldots, m-1\}$
end

Figure 1: SVRG algorithm

Answer
Use Option II. By telescoping, we have

$$\frac{1}{m}\mathbb{E}[\|w_m - w^*\|^2 - \|w_0 - w^*\|^2] \leq -2\eta(1-2L\eta)\left(\frac{1}{m}\sum_{i=0}^{m-1} f(i) - f(w^*)\right) + 4L\eta^2(f(\tilde{w}) - f(w^*))$$

Since $\frac{1}{m}\sum_{i=0}^{m-1} f(i) = f(\tilde{w}_s)$ by definition, we have

$$2\eta(1-2L\eta)\mathbb{E}[f(\tilde{w}_s) - f(w^*)] \leq \frac{1}{m}\mathbb{E}\|w_0 - w^*\|^2 + 4L\eta^2\mathbb{E}[f(\tilde{w}) - f(w^*)].$$

By the strong convexity, we have $f(w_0) \geq f(w^*) + \frac{\mu}{2}\|w_0 - w^*\|^2$, so

$$2\eta(1-2L\eta)\mathbb{E}[f(\tilde{w}_s) - f(w^*)] \leq \frac{2}{\mu m}(f(w_0) - f(w^*)) + 4L\eta^2\mathbb{E}[f(\tilde{w}) - f(w^*)].$$

Dividing both sides by $2\eta(1-2L\eta)$ and using that $w_0 = \tilde{w}_{s-1} = \tilde{w}$ would prove the desired result.