Homework #4

RED CORRECTION: 04/23/2023 13:30

RELEASE DATE: 04/20/2023

DUE DATE: 05/11/2023, BEFORE 13:00 on Gradescope

QUESTIONS ARE WELCOMED ON DISCORD.

You will use Gradescope to upload your choices and your scanned/printed solutions. For problems marked with (*), please follow the guidelines on the course website and upload your source code to Gradescope as well. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 20 problems and a total of 500 points. For Problems 12-20, there is one correct choice. If you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get 0 points. For eight of the secretly-selected problems out of Problems 1-11, the TAs will grade your detailed solution in terms of the written explanations and/or code based on how logical/clear your solution is. Each of the eight problems graded by the TAs counts as 40 points (in addition to the correct/incorrect choices you made). In general, each homework (except homework 0) is of a total of 500 points.

More about Regularization

1. Consider a one-dimensional data set $\{(x_n, y_n)\}_{n=1}^N$ where each $x_n \in \mathbb{R}$ and $y_n \in \mathbb{R}$. Then, solve the following one-variable regularized linear regression problem:

$$\min_{w\in\mathbb{R}} \frac{1}{N}\sum_{n=1}^N (w \cdot x_n - y_n)^2 + \frac{\lambda}{N} w^2.$$

If the optimal solution to the problem above is w^*, it can be shown that w^* is also the optimal solution of

$$\min_{w\in\mathbb{R}} \frac{1}{N}\sum_{n=1}^N (w \cdot x_n - y_n)^2 \text{ subject to } w^2 \le C$$

with $C = (w^*)^2$. This allows us to express the relationship between C in the constrained optimization problem and λ in the augmented optimization problem for any $\lambda > 0$. What is the relationship? Choose the correct answer; explain your answer.

[a] $C = \left(\dfrac{\sum_{n=1}^N x_n y_n}{\sum_{n=1}^N x_n^2 + \lambda}\right)^2$ [*] ....

[b] $C = \left(\dfrac{\sum_{n=1}^N y_n^2}{\sum_{n=1}^N x_n^2 + \lambda}\right)^2$

[c] $C = \left(\dfrac{\sum_{n=1}^N x_n^2 y_n^2}{\sum_{n=1}^N x_n^2 + \lambda}\right)^2$

[d] $C = \left(\dfrac{\sum_{n=1}^N x_n y_n}{\sum_{n=1}^N y_n^2 + \lambda}\right)^2$

[e] $C = \left(\dfrac{\sum_{n=1}^N x_n^2}{\sum_{n=1}^N y_n^2 + \lambda}\right)^2$

(*Note: All the choices hint you that a smaller λ corresponds to a bigger C.*)

2. The ranges of features may affect regularization. One common technique to align the ranges of features is to consider a "normalization" transformation. Define $\mathbf{\Phi}(\mathbf{x}) = \Gamma^{-1}(\mathbf{x} - \mathbf{u})$, where $\mathbf{u}$ is an estimated mean of the examples, Γ is a diagonal matrix with positive diagonal values $\gamma_0, \gamma_1, \ldots, \gamma_d$ that indicate the estimated standard deviation. For simplicity, consider $\mathbf{u} = \mathbf{0}$. Then, conducting L2-regularized linear regression in the $\mathcal{Z}$-space

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (\tilde{\mathbf{w}}^T \mathbf{\Phi}(\mathbf{x}_n) - y_n)^2 + \frac{\lambda}{N}(\tilde{\mathbf{w}}^T \tilde{\mathbf{w}})$$

is equivalent to regularized linear regression in the $\mathcal{X}$-space

$$\min_{\mathbf{w} \in \mathbb{R}^{d+1}} \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{x}_n - y_n)^2 + \frac{\lambda}{N}\Omega(\mathbf{w})$$

with a different regularizer $\Omega(\mathbf{w})$. What is $\Omega(\mathbf{w})$? Choose the correct answer; explain your answer.

[a] $\mathbf{w}^T \Gamma \mathbf{w}$

[b] $\mathbf{w}^T \Gamma^2 \mathbf{w}$ [*] ....

[c] $\mathbf{w}^T \mathbf{w}$

[d] $\mathbf{w}^T \Gamma^{-2} \mathbf{w}$

[e] $\mathbf{w}^T \Gamma^{-1} \mathbf{w}$

3. The error function of logistic regression

$$\text{err}(\mathbf{w}, \mathbf{x}, y) = \ln(1 + \exp(-y\mathbf{w}^T\mathbf{x}))$$

can be re-written as

$$\text{err}(\mathbf{w}, \mathbf{x}, y) = [\![y = +1]\!] \ln(1 + \exp(-\mathbf{w}^T\mathbf{x})) + [\![y = -1]\!] \ln(1 + \exp(\mathbf{w}^T\mathbf{x})).$$

Label smoothing is a popular way of combatting overfitting by replacing the error function with a smoothed one

$$\text{err}_{smooth}(\mathbf{w}, \mathbf{x}, +1) = (1 - \frac{\alpha}{2}) \ln(1 + \exp(-\mathbf{w}^T\mathbf{x})) + \frac{\alpha}{2} \ln(1 + \exp(\mathbf{w}^T\mathbf{x})).$$

and

$$\text{err}_{smooth}(\mathbf{w}, \mathbf{x}, -1) = \frac{\alpha}{2} \ln(1 + \exp(-\mathbf{w}^T\mathbf{x})) + (1 - \frac{\alpha}{2}) \ln(1 + \exp(\mathbf{w}^T\mathbf{x})).$$

Solving the in-sample error using the smoothed error function

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \text{err}_{smooth}(\mathbf{w}, \mathbf{x}_n, y_n)$$

is equivalent to solving a regularized logistic regression problem.

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} \text{err}(\mathbf{w}, \mathbf{x}_n, y_n) + \frac{\lambda}{N} \sum_{n=1}^{N} \Omega(\mathbf{w}, \mathbf{x}_n).$$

Let $D_{KL}(P||Q)$ denote the KL-divergence between two probability distributions P and Q and let $P_u(+1) = P_u(-1) = \frac{1}{2}$ denote a uniform probability distribution on binary outcomes. Note that every logistic hypothesis $h(\mathbf{x})$ defines a probability distribution $P_h(+1|\mathbf{x}) = h(\mathbf{x})$ and $P_h(-1|\mathbf{x}) = (1 - h(\mathbf{x}))$. Let $\lambda = \frac{\alpha}{1-\alpha}$. What is $\Omega(\mathbf{w}, \mathbf{x})$? Choose the correct answer; explain your answer.

[a] $D_{KL}(P_u||P_h)$ [*] ....

[b] $D_{KL}(P_h||P_u)$

[c] $\frac{1}{2}(D_{KL}(P_u||P_h) + D_{KL}(P_h||P_u))$

[d] $D_{KL}(P_u||P_h) + D_{KL}(P_h||P_u)$

[e] none of the other choices

Validation

4. Consider three examples (x_1, y_1), (x_2, y_2), $(x_3, y_3 = 1)$. Assume that x_1, x_2, x_3 are independent random variables that are uniformly generated between $[-1, 1]$, and y_1, y_2 are independent random variables that are uniformly generated between $[0, 2]$. Use leave-one-out cross-validation with the squared error to estimate the performance of the constant model, which returns the best constant hypothesis $h(x) = w_0$ in terms of the squared error. What is the probability that $E_{loocv} \le \frac{1}{3}$? Choose the correct answer; explain your choice.

[a] $\dfrac{\pi}{12}$

[b] $\dfrac{\pi}{3\sqrt{3}}$ [*]

[c] $\dfrac{\pi}{2\sqrt{6}}$

[d] $\dfrac{\pi}{2\sqrt{3}}$

[e] none of the other choices

5. Consider a probability distribution $\mathcal{P}(\mathbf{x}, y)$ that can be used to generate examples $(\mathbf{x}, y)$, and suppose we generate K i.i.d. examples from the distribution as validation examples, and store them in $\mathcal{D}_{\text{val}}$. For any fixed hypothesis h, we can show that

$$\underset{\mathcal{D}_{\text{val}} \sim \mathcal{P}^K}{\text{Variance}}\big[E_{\text{val}}(h)\big] = \square \cdot \underset{(\mathbf{x},y) \sim \mathcal{P}}{\text{Variance}}\big[\text{err}(h(\mathbf{x}), y)\big].$$

Which of the following is $\square$? Choose the correct answer; explain your answer.

[a] K

[b] $\sqrt{K}$

[c] $\frac{1}{\sqrt{K}}$

[d] $\frac{1}{K}$ [*] ....

[e] none of the other choices

6. Consider a binary classification algorithm $\mathcal{A}_{\text{majority}}$, which returns a constant classifier that always predicts the majority class (i.e., the class with more instances in the data set that it sees). As you can imagine, the returned classifier is the best-E_{in} one among all constant classifiers. For a binary classification data set with N positive examples and N negative examples, what is $E_{\text{loocv}}(\mathcal{A}_{\text{majority}})$? Choose the correct answer; explain your answer.

[a] $1/(N-1)$

[b] $1/N$

[c] $1/(N+1)$

[d] 1 [*] ....

[e] none of the other choices

7. Consider the decision stump model and the data generation process of generate x by a uniform distribution in $[-1, +1]$ and $y = \text{sign}(x)$. Use the generation process to generate a data set of N examples (instead of 2). If the data set contains at least two positive examples and at least two negative examples, which of the following is the tightest upper bound on the leave-one-out error of the decision stump model? Choose the correct answer; explain your answer.

[a] 0

[b] $1/N$

[c] $2/N$ [*] ....

[d] $1/2$

[e] 1

Support Vector Machine

8. Consider N "linearly separable" 1D examples $\{(x_n, y_n)\}_{n=1}^N$. That is, $x_n \in \mathbb{R}$. Without loss of generality, assume that $x_1 \le x_2 \le \ldots x_M < x_{M+1} \le x_{M+2} \ldots \le x_N$, $y_n = -1$ for $n = 1, 2, \ldots, M$, and $y_n = +1$ for $n = M+1, M+2, \ldots, N$. Apply hard-margin SVM without transform on this data set. What is the largest margin achieved? Choose the correct answer; explain your answer.

[a] $\frac{1}{2}(x_N - x_M)$

[b] $\frac{1}{2}(x_{M+1} - x_1)$

[c] $\frac{1}{2}\left(\frac{1}{N-M}\sum_{n=M+1}^{N} x_n - \frac{1}{M}\sum_{n=1}^{M} x_n\right)$

[d] $\frac{1}{2}(x_N - x_1)$

[e] $\frac{1}{2}(x_{M+1} - x_M)$ [*] ....

(*Hint: Have we mentioned that a decision stump is just a 1D perceptron, and the hard-margin SVM is an extension of the perceptron model? :-)*)

9. In some situations, we expect to achieve a smaller margin for the positive examples and a larger margin for the negative examples. For instance, when there are very few negative examples and a lot more positive examples, giving the nagaive examples a smaller margin could be more robust. Consider an *uneven-margin* support vector machine that solves

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & (\mathbf{w}^T\mathbf{x}_n + b) \ge 1 \text{ for } y_n = +1 \\ & -(\mathbf{w}^T\mathbf{x}_n + b) \ge 1126 \text{ for } y_n = -1. \end{aligned}$$

Given the following examples.

$\mathbf{x}_1 = (0, 4)$	$y_1 = +1$
$\mathbf{x}_2 = (2, 0)$	$y_2 = -1$
$\mathbf{x}_3 = (-1, 0)$	$y_3 = +1$
$\mathbf{x}_4 = (0, 0)$	$y_4 = +1$

What is the optimal $\mathbf{w}$ and b? Choose the correct answer; explain your answer.

[a] the optimal $\mathbf{w} = (\frac{-1127}{3}, 0), b = 1$

[b] the optimal $\mathbf{w} = (\frac{-1125}{2}, 0), b = -1$

[c] the optimal $\mathbf{w} = (\frac{-1125}{3}, 0), b = -1$

[d] the optimal $\mathbf{w} = (0, \frac{1127}{4}), b = 1$

[e] the optimal $\mathbf{w} = (\frac{-1127}{2}, 0), b = 1$ [*] ....

10. For a set of examples $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ and a kernel function K, consider a hypothesis set that contains

$$h_{\boldsymbol{\alpha},b}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} y_n \alpha_n K(\mathbf{x}_n, \mathbf{x}) + b\right).$$

The classifier returned by SVM can be viewed as one such $h_{\boldsymbol{\alpha},b}$, where the values of $\boldsymbol{\alpha}$ is determined by the dual QP solver and b is calculated from the KKT conditions.

In this problem, we study a simpler form of $h_{\boldsymbol{\alpha},b}$ where $\boldsymbol{\alpha} = \mathbf{1}$ (the vector of all 1's) and $b = 0$. Let us name $h_{\mathbf{1},0}$ as $\hat{h}$ for simplicity. We will show that when using the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$, if γ is large enough, $E_{\text{in}}(\hat{h}) = 0$. That is, when using the Gaussian kernel, we can "easily" separate the given data set if γ is large enough.

Assume that the distance between any pair of different $(\mathbf{x}_n, \mathbf{x}_m)$ in the $\mathcal{X}$-space is no less than ϵ. That is,

$$\|\mathbf{x}_n - \mathbf{x}_m\| \geq \epsilon \quad \forall n \neq m.$$

What is the tightest lower bound of γ that ensures $E_{\text{in}}(\hat{h}) = 0$? Choose the correct answer; explain your answer.

[a] $\frac{\ln^2(N+1)}{\epsilon^2}$

[b] $\frac{\ln(N+1)}{\epsilon^2}$

[c] $\frac{\ln(N)}{\epsilon^2}$

[d] $\frac{\ln(N-1)}{\epsilon^2}$ [*] ....

[e] $\frac{\ln^2(N-1)}{\epsilon^2}$

11. For any feature transform $\boldsymbol{\phi}$ from $\mathcal{X}$ to $\mathcal{Z}$, the squared distance between two examples $\mathbf{x}$ and $\mathbf{x}'$ is $\|\boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\phi}(\mathbf{x}')\|^2$ in the $\mathcal{Z}$-space. For the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$, compute the distance with the kernel trick. Then, for any two examples $\mathbf{x}$ and $\mathbf{x}'$, among the choices, what is the tightest upper bound for their distance in the $\mathcal{Z}$-space? Choose the correct answer; explain your answer.

[a] 0.0

[b] 0.5

[c] 1.0

[d] 1.5 [*] ....

[e] 2.0

Experiments with Regularized Logistic Regression

Consider L2-regularized logistic regression with fourth-order polynomial transformation.

$$\mathbf{w}_\lambda = \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{N}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{n=1}^{N} \ln(1+\exp(-y_n \mathbf{w}^T \mathbf{\Phi}_4(\mathbf{x}_n))),$$

Here $\mathbf{\Phi}_4$ is the fourth-order polynomial transformation introduced in our lecture (with $Q = 4$), defined as

$$\mathbf{\Phi}_4(\mathbf{x}) = (1, x_1, x_2, \ldots, x_d, x_1^2, x_1x_2, \ldots, x_d^2, \ldots, x_1^4, x_1^3x_2, \ldots, x_d^4)$$

Next, we will take the following file as our training data set $\mathcal{D}$:

`http://www.csie.ntu.edu.tw/~htlin/course/ml23fall/hw4/hw4_train.dat`

and the following file as our test data set for evaluating E_{out}:

`http://www.csie.ntu.edu.tw/~htlin/course/ml23fall/hw4/hw4_test.dat`

We call the algorithm for solving the problem above as $\mathcal{A}_\lambda$. The problem guides you to use LIBLINEAR (`https://www.csie.ntu.edu.tw/~cjlin/liblinear/`), a machine learning package developed in our university, to solve this problem. In addition to using the default options, what you need to do when running LIBLINEAR are

- set option `-s 0`, which corresponds to solving L2-regularized logistic regression
- set option `-c C`, with a parameter value of `C` calculated from the λ that you want to use; read `README` of the software package to figure out how `C` and your λ should relate to each other
- set option `-e 0.000001`, which corresponds to getting a solution that is really really close to the optimal solution

LIBLINEAR can be called from the command line or from major programming languages like python. If you run LIBLINEAR in the command line, please include screenshots of your scripts/commands/results; if you run LIBLINEAR from any programming language, please include screenshots of your code.

We will consider the data set as a *binary classification problem* and take the "regression for classification" approach with regularized logistic regression. So please evaluate all errors below with the 0/1 error.

Hint: Be sure to double-check the following steps.

- Verify the dimension of $\mathbf{\Phi}_4$.
- Verify the relationship between λ and C. Be careful of $\log_{10}\lambda$ in the problem.
- Verify the command of LIBLINEAR.
- Use 0/1 *error* rather than *accuracy*.

12. (*) Select the best λ^* *in a cheating manner* as

$$\operatorname*{argmin}_{\log_{10}\lambda \in \{-6,-3,0,3,6\}} E_{\text{out}}(\mathbf{w}_\lambda).$$

Break the tie, if any, by selecting the largest λ. What is $\log_{10}(\lambda^*)$? Choose the closest answer; provide your command/code.

[a] -6

[b] -3

[c] 0

[d] 3

[e] 6

13. (*) Select the best λ^* as

$$\underset{\log_{10}\lambda\in\{-6,-3,0,3,6\}}{\operatorname{argmin}} E_{\text{in}}(\mathbf{w}_\lambda).$$

Break the tie, if any, by selecting the largest λ. What is $\log_{10}(\lambda^*)$? Choose the closest answer; provide your command/code.

[a] -6

[b] -3

[c] 0

[d] 3

[e] 6

14. (*) Now randomly split the given training examples in $\mathcal{D}$ to two sets: 120 examples as $\mathcal{D}_{\text{train}}$ and 80 as $\mathcal{D}_{\text{val}}$. Run $\mathcal{A}_\lambda$ on *only* $\mathcal{D}_{\text{train}}$ to get $\mathbf{w}_\lambda^-$ (the weight vector within the g^- returned), and validate $\mathbf{w}_\lambda^-$ with $\mathcal{D}_{\text{val}}$ to get $E_{\text{val}}(\mathbf{w}_\lambda^-)$. Select the best λ^* as

$$\underset{\log_{10}\lambda\in\{-6,-3,0,3,6\}}{\operatorname{argmin}} E_{\text{val}}(\mathbf{w}_\lambda^-).$$

Break the tie, if any, by selecting the largest λ. Repeat the experiment for 256 times, each with a different random split. What is the λ that is selected the most often? Choose the closest answer; provide your command/code.

[a] -6

[b] -3

[c] 0

[d] 3

[e] 6

15. (*) Repeat the 256 experiments in the previous problem, and estimate $E_{\text{out}}(\mathbf{w}_{\lambda^*}^-)$ with the test set in each round of the experiments. What is the average value of $E_{\text{out}}(\mathbf{w}_{\lambda^*}^-)$? Choose the closest answer; provide your command/code.

[a] 0.13

[b] 0.15

[c] 0.17

[d] 0.19

[e] 0.21

16. (*) Repeat the 256 experiments in the previous problem, but run $\mathcal{A}_\lambda$ on *the full* $\mathcal{D}$ to get $\mathbf{w}_\lambda$ instead. Then, estimate $E_{\text{out}}(\mathbf{w}_{\lambda^*})$ with the test set. What is the average value of $E_{\text{out}}(\mathbf{w}_{\lambda^*})$? Choose the closest answer; provide your command/code.

[a] 0.13

[b] 0.15

[c] 0.17

[d] 0.19

[e] 0.21

17. Now randomly split the given training examples in $\mathcal{D}$ to five folds, the first 40 being fold 1, the next 40 being fold 2, and so on. Select the best λ^* as

$$\mathop{\text{argmin}}_{\log_{10}\lambda\in\{-6,-3,0,3,6\}} E_{\text{cv}}(\mathcal{A}_\lambda).$$

Break the tie, if any, by selecting the largest λ. Repeat the experiment for 256 times. What is the average value of $E_{\text{cv}}(\mathcal{A}_{\lambda^*})$ Choose the closest answer; provide your command/code.

[a] 0.13

[b] 0.15

[c] 0.17

[d] 0.19

[e] 0.21

Next, consider L1-regularized logistic regression with second-order polynomial transformation.

$$\mathbf{w}_\lambda = \mathop{\text{argmin}}_{\mathbf{w}} \frac{\lambda}{N}\|\mathbf{w}\|_1 + \frac{1}{N}\sum_{n=1}^{N}\ln(1+\exp(-y_n\mathbf{w}^T\mathbf{\Phi}_4(\mathbf{x}_n))),$$

In addition to using the default options, what you need to do when running LIBLINEAR are

- set option `-s 6`, which corresponds to solving L1-regularized logistic regression
- set option `-c C`, with a parameter value of `C` calculated from the λ that you want to use; read `README` of the software package to figure out how `C` and your λ should relate to each other
- set option `-e 0.000001`, which corresponds to getting a solution that is really really close to the optimal solution

18. (*) For L1-regularized logistic regression, select the best λ^* *in a cheating manner* as

$$\mathop{\text{argmin}}_{\log_{10}\lambda\in\{-6,-3,0,3,6\}} E_{\text{out}}(\mathbf{w}_\lambda).$$

Break the tie, if any, by selecting the largest λ. What is $\log_{10}(\lambda^*)$? Choose the closest answer; provide your command/code.

[a] -6

[b] -3

[c] 0

[d] 3

[e] 6

19. (*) Based on the λ^* chosen in the previous problem, obtain $\mathbf{w}_{\lambda^*}$ from L1-regularized logistic regression. How sparse is $\mathbf{w}_{\lambda^*}$? That is, how many components w_i within $\mathbf{w}_{\lambda^*}$ satisfies $|w_i| \le 10^{-6}$? Choose the closest answer; provide your command/code.

[a] 1

[b] 200

[c] 400

[d] 800

[e] 1000

20. (*) Based on the λ^* chosen in the Problem 12, obtain $\mathbf{w}_{\lambda^*}$ from **L2-regularized** logistic regression. How sparse is $\mathbf{w}_{\lambda^*}$? That is, how many components w_i within $\mathbf{w}_{\lambda^*}$ satisfies $|w_i| \le 10^{-6}$? Choose the closest answer; provide your command/code.

[a] 1

[b] 200

[c] 400

[d] 800

[e] 1000