Homework #5

RED RELEASE DATE WHICH INCLUDES P17-P20: 05/05/2023

RELEASE DATE: 05/04/2023

DUE DATE: 06/01/2023, 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 each problem, 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 five of the secretly-selected problems, 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 five problems graded by the TAs counts as additional 20 points (in addition to the correct/incorrect choices you made). In general, each homework (except homework 0) is of a total of 500 points.

Support Vector Machines

1. The soft-margin support vector machine tolerates some errors through the so-called slack variables ξ_n.

$$\begin{array}{lll} \min\limits_{\mathbf{w},b,\xi_n} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum\limits_{n=1}^{N}\xi_n & \\ \text{subject to} & y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n & , \text{for } n = 1, \ldots, N \\ & \xi_n \geq 0 & , \text{for } n = 1, \ldots, N. \end{array}$$

After solving the soft-margin support vector machine and obtaining the optimal $(b^*, \mathbf{w}^*)$ and $\xi_1^*, \ldots, \xi_N^*$, how many of the following terms are upper bounds of the number of misclassified examples?

- $\sum_{n=1}^{N} \frac{\xi_n^*}{2}$
- $\sum_{n=1}^{N} \sqrt{\xi_n^*}$
- $\sum_{n=1}^{N} \xi_n^*$
- $\sum_{n=1}^{N} \lfloor \xi_n^* \rfloor$
- $\sum_{n=1}^{N} \log_2(1 + \xi_n^*)$

Choose the correct answer; briefly explain the formula that correspond to your choice.

[a] 1

[b] 2

[c] 3

[d] 4

[e] 5

2. Consider the soft-margin support vector machine taught in our class. Assume that after solving the dual problem, every example is a bounded support vector. That is, the optimal solution $\boldsymbol{\alpha}^*$ satisfies $\alpha_n^* = C$ for every example. In this case, there may be multiple solutions for the optimal b^* for the primal support vector machine problem. What is the smallest such b^*? Choose the correct answer; prove your choice.

[a] $\min\limits_{n:\, y_n > 0} \left(1 - \sum_{m=1}^{N} y_m \alpha_m^* K(\mathbf{x}_n, \mathbf{x}_m)\right)$

[b] $\max\limits_{n:\, y_n > 0} \left(1 - \sum_{m=1}^{N} y_m \alpha_m^* K(\mathbf{x}_n, \mathbf{x}_m)\right)$

[c] $\min\limits_{n:\, y_n < 0} \left(-1 - \sum_{m=1}^{N} y_m \alpha_m^* K(\mathbf{x}_n, \mathbf{x}_m)\right)$

[d] $\max\limits_{n:\, y_n < 0} \left(-1 - \sum_{m=1}^{N} y_m \alpha_m^* K(\mathbf{x}_n, \mathbf{x}_m)\right)$

[e] none of the other choices

3. Consider the non-linear soft-margin support vector machine that we have taught in class.

$$\begin{aligned}(P_1) \quad \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\ \text{subject to} \quad & y_n\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) + b\right) \geq 1 - \xi_n, \text{ for } n = 1, 2, \ldots, N, \\ & \xi_n \geq 0, \text{ for } n = 1, 2, \ldots, N.\end{aligned}$$

The support vector machine penalizes the margin violations linearly. Another popular formulation penalizes the margin violations quadratically. The formulation is as follows:

$$\begin{aligned}(P_2) \quad \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n^2 \\ \text{subject to} \quad & y_n\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) + b\right) \geq 1 - \xi_n, \text{ for } n = 1, 2, \ldots, N.\end{aligned}$$

Note that we do not have the $\xi_n \geq 0$ constraints as any negative ξ_n would never be an optimal solution of (P_2)—you are encouraged to think about why. Anyway, the dual problem of (P_2) will look like this:

$$\begin{aligned}(D_2) \quad \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \cdot \left(K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{2C}[\![n = m]\!]\right) - \sum_{n=1}^{N}\alpha_n \\ \text{subject to} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0 \\ & \alpha_n \geq 0, \text{ for } n = 1, 2, \ldots, N,\end{aligned}$$

where the kernel function $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T\boldsymbol{\phi}(\mathbf{x}')$. After getting the optimal $\boldsymbol{\alpha}^*$ for (D_2), how can we calculate the optimal $\boldsymbol{\xi}^*$ for (P_2)? Choose the correct answer; prove your choice.

[a] $\boldsymbol{\xi}^* = \frac{1}{2C}\boldsymbol{\alpha}^*$

[b] $\boldsymbol{\xi}^* = \frac{1}{2}\boldsymbol{\alpha}^*$

[c] $\boldsymbol{\xi}^* = \frac{1}{C}\boldsymbol{\alpha}^*$

[d] $\boldsymbol{\xi}^* = C\boldsymbol{\alpha}^*$

[e] none of the other choices

(*Note: It is very interesting that* (D_2) *looks like a hard-margin support vector machine, where* α_n*'s are not upper-bounded. You are encouraged to think about why.*)

4. When talking about non-uniform voting in aggregation, we mentioned that $\boldsymbol{\alpha}$ can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

$$\boldsymbol{\phi}(\mathbf{x}) = \Big(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_T(\mathbf{x})\Big).$$

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product $(\boldsymbol{\phi}(\mathbf{x}))^T(\boldsymbol{\phi}(\mathbf{x}'))$. In this problem, we mix the two topics together using the decision stumps as our $g_t(\mathbf{x})$.

Assume that the input vectors contain only even integers between (including) $2L$ and $2R$, where $L < R$. Consider the decision stumps $g_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}\Big(x_i - \theta\Big)$, where

$$\begin{aligned} i &\quad \in \{1, 2, \cdots, d\}, \\ d &\quad \text{is the finite dimensionality of the input space,} \\ s &\quad \in \{-1, +1\}, \\ \theta &\quad \text{is an odd integer between } (2L, 2R). \end{aligned}$$

Define $\boldsymbol{\phi}_{ds}(\mathbf{x}) = \Bigg(g_{+1,1,2L+1}(\mathbf{x}), g_{+1,1,2L+3}(\mathbf{x}), \ldots, g_{+1,1,2R-1}(\mathbf{x}), \ldots, g_{-1,d,2R-1}(\mathbf{x})\Bigg)$. What is $K_{ds}(\mathbf{x}, \mathbf{x}') = (\boldsymbol{\phi}_{ds}(\mathbf{x}))^T(\boldsymbol{\phi}_{ds}(\mathbf{x}'))$? Choose the correct answer; explain your choice.

[a] $2d(R - L) - \|\mathbf{x} - \mathbf{x}'\|_1$

[b] $4d(R - L) - \|\mathbf{x} - \mathbf{x}'\|_2$

[c] $2d(R - L)^2 - \|\mathbf{x} - \mathbf{x}'\|_1^2$

[d] $4d(R - L)^2 - \|\mathbf{x} - \mathbf{x}'\|_2^2$

[e] none of the other choices

Bagging and Boosting

5. Consider an aggregated binary classifier G that is constructed by uniform blending on $2M + 1$ binary classifiers $\{g_t\}_{t=1}^{2M+1}$. That is,

$$G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{2M+1} g_t(\mathbf{x})\right)$$

Assume that each g_t is of test 0/1 error $E_{\text{out}}(g_t) = e_t$. Which of the following is the tightest upper bound of $E_{out}(G)$? Choose the correct answer; prove your choice.

[a] $\frac{1}{M} \sum_{t=1}^{2M+1} e_t$

[b] $\frac{1}{M+1} \sum_{t=1}^{2M+1} e_t$

[c] $\frac{1}{2M} \sum_{t=1}^{2M+1} e_t$

[d] $\frac{1}{2M+1} \sum_{t=1}^{2M+1} e_t$

[e] none of the other choices

6. Suppose we have a data set of size $N = 1127$, and we use bootstrapping to sample N' examples. What is the smallest N' such that the probability of getting at least one duplicated example (i.e. some $(\mathbf{x}_n, y_n)$ being selected more than once) is larger than 75%? Choose the correct answer; explain your choice.

[a] 54

[b] 56

[c] 58

[d] 60

[e] none of the other choices

7. Assume that linear regression (for classification) is used within AdaBoost. That is, we need to solve the weighted-E_{in} optimization problem for some given $u_n \geq 0$.

$$\min_{\mathbf{w}} E_{\text{in}}^{\mathbf{u}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} u_n (y_n - \mathbf{w}^T\mathbf{x}_n)^2.$$

The optimization problem above is equivalent to minimizing the usual E_{in} of linear regression on some "pseudo data set" $\{(\tilde{\mathbf{x}}_n, \tilde{y}_n)\}_{n=1}^N$. Which of the following denotes an equivalent pseudo data set? Choose the correct answer; explain your choice.

[a] $\tilde{\mathbf{x}}_n = \mathbf{x}_n, \tilde{y}_n = u_n y_n$

[b] $\tilde{\mathbf{x}}_n = u_n \mathbf{x}_n, \tilde{y}_n = y_n$

[c] $\tilde{\mathbf{x}}_n = \sqrt{u_n}\mathbf{x}_n, \tilde{y}_n = \sqrt{u_n} y_n$

[d] $\tilde{\mathbf{x}}_n = \frac{1}{\sqrt{u_n}}\mathbf{x}_n, \tilde{y}_n = \frac{1}{\sqrt{u_n}} y_n$

[e] none of the other choices

8. When evaluated with the Gini index as the impurity criteria, which of the following choices is the best branch for building a CART decision tree? Choose the correct answer; explain your choice.

[a] a 50/50 split, with the first part containing all positive examples, and the second part containing 50% positive examples (i.e. 50% negative).

[b] a 70/30 split, with the first part containing 80% positive examples (20% negative) examples, and the second part containing 75% positive examples (25% negative).

[c] a 90/10 split, with the first part containing 70% positive examples (30% negative) examples, and the second part containing all negative examples.

[d] a 80/20 split, with the first part containing 80% positive examples (20% negative) examples, and the second part containing 90% positive examples (10% negative) examples.

[e] a 80/20 split, with the first part containing 90% positive examples (10% negative) examples, and the second part containing 90% negative examples (10% positive) examples.

9. For the AdaBoost algorithm introduced in class, let $U_t = \sum_{n=1}^{N} u_n^{(t)}$. Note that $U_1 = \sum_{n=1}^{N} \frac{1}{N} = 1$. Assume that $0 < \epsilon_t < \frac{1}{2}$ for each hypothesis g_t. After the algorithm runs for T iterations, what is U_{T+1}? Choose the correct answer; prove your choice.

[a] $\sum_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$

[b] $2\sum_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$

[c] $\prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$

[d] $2^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$

[e] none of the other choices

10. For the gradient boosted decision tree algorithm introduced in class, after updating all s_n in iteration t using the steepest η as α_t, what is the value of

$$\sum_{n=1}^{N} (s_n - y_n) g_t(\mathbf{x}_n)$$

when using the new (updated) s_n? Choose the correct answer; prove your choice.

[a] $-\sum_{n=1}^{N} (g_t(\mathbf{x}_n))^2$

[b] 0

[c] $+\sum_{n=1}^{N} |g_t(\mathbf{x}_n)|$

[d] $+\sum_{n=1}^{N} (g_t(\mathbf{x}_n))^2$

[e] none of the other choices

Experiments with Soft-Margin SVM and AdaBoost

For the problems in our last homework set (yeah!!), we are going to experiment with a real-world data set. Download the letter data sets from LIBSVM Tools.

Training: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/letter.scale.tr

Testing: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/letter.scale.t

We will consider binary classification problems of the form "one of the classes" (as the positive class) versus "the other classes" (as the negative class). That is, they are one sub-problems of the one-versus-all decomposition that we have taught.

Problems 11-16 will surround soft-margin support vector machines. The data set contains thousands of examples, and some quadratic programming packages cannot handle this size. We recommend that you consider the LIBSVM package

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

LIBSVM can be called from the command line or from major programming languages like python. If you run LIBSVM in the command line, please include screenshots of your scripts/commands/results; if you run LIBSVM from any programming language, please include screenshots of your code. If you choose not to use LIBSVM, please include scripts/commands/results/screenshots of your chosen solver similarly.

Regardless of the package that you choose to use, please read the manual of the package carefully to make sure that you are indeed solving the soft-margin support vector machine taught in class like the dual formulation below:

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) - \sum_{n=1}^{N}\alpha_n \\
\text{subjectto} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0 \\
& 0 \le \alpha_n \le C \quad n = 1, \ldots, N.
\end{aligned}$$

In the following problems, please use the 0/1 error for evaluating E_{in}, E_{val} and E_{out} (through the test set). Some practical remarks include

(i) Please tell your chosen package to **not** automatically scale the data for you, lest you should change the effective kernel and get different results.

(ii) It is your responsibility to check whether your chosen package solves the designated formulation with enough numerical precision. Please read the manual of your chosen package for software parameters whose values affect the outcome—any ML practitioner needs to deal with this kind of added uncertainty.

11. (*) Consider the linear soft-margin SVM. That is, either solve the primal formulation of soft-margin SVM with the given $\mathbf{x}_n$, or take the linear kernel $K(\mathbf{x}_n, \mathbf{x}_m) = \mathbf{x}_n^T \mathbf{x}_m$ in the dual formulation. With $C = 1$, and the binary classification problem of "1" versus "not 1", which of the following numbers is closest to $\|\mathbf{w}\|$ after solving the linear soft-margin SVM? Choose the closest answer; provide your command/code.

[**a**] 4.31

[**b**] 5.31

[**c**] 6.31

[**d**] 7.31

[**e**] 8.31

12. (*) Consider the polynomial kernel $K(\mathbf{x}_n, \mathbf{x}_m) = (1 + \mathbf{x}_n^T \mathbf{x}_m)^Q$, where Q is the degree of the polynomial. With $C = 1$, $Q = 2$, which of the following soft-margin SVM classifiers reaches the largest E_{in}? Choose the correct answer; provide your command/code.

[**a**] "2" versus "not 2"

[**b**] "3" versus "not 3"

[**c**] "4" versus "not 4"

[**d**] "5" versus "not 5"

[**e**] "6" versus "not 6"

13. (*) Following Problem 12, which of the following numbers is closest to the minimum number of support vectors within those five soft-margin SVM classifiers? Choose the closest answer; provide your command/code.

[**a**] 250

[**b**] 350

[**c**] 450

[**d**] 550

[**e**] 650

14. (*) Consider the Gaussian kernel $K(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\gamma ||\mathbf{x}_n - \mathbf{x}_m||^2\right)$. For the binary classification problem of "7" versus "not 7", when fixing $\gamma = 1$, which of the following values of C results in the lowest E_{out}? If there is a tie, please pick the smallest C. Choose the correct answer; provide your command/code.

[**a**] 0.01

[**b**] 0.1

[**c**] 1

[**d**] 10

[**e**] 100

15. (*) Following Problem 14, when fixing $C = 0.1$, which of the following values of γ results in the lowest E_{out}? If there is a tie, please pick the smallest γ. Choose the correct answer; provide your command/code.

- [a] 0.1
- [b] 1
- [c] 10
- [d] 100
- [e] 1000

16. (*) Following Problem 14 and consider a validation procedure that randomly samples 200 examples from the training set for validation and leaves the other examples for training g^{-}_{SVM}. Fix $C = 0.1$ and use the validation procedure to choose the best γ among $\{0.1, 1, 10, 100, 1000\}$ according to E_{val}. If there is a tie of E_{val}, choose the smallest γ. Repeat the procedure 500 times. Which of the following values of γ is selected the most number of times? Choose the correct answer; provide your command/code.

- [a] 0.1
- [b] 1
- [c] 10
- [d] 100
- [e] 1000

For Problems 17-20, implement the AdaBoost-Stump algorithm as introduced in class. Run the algorithm on the letter data for the **one-versus-one** binary classification problem of label "11" versus label "26":

Training: `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/letter.scale.tr`

Testing: `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/letter.scale.t`

Please use a total of $T = 1000$ iterations (please do not stop earlier than 1000), and calculate E_{in} and E_{out} with the 0/1 error. You can take $\text{sign}(0) = +1$ if needed.

For the decision stump algorithm within AdaBoost-Stump, please implement the following steps. Any ties can be arbitrarily broken.

(1) For any feature i, sort all the $x_{n,i}$ values to $x_{[n],i}$ such that $x_{[n],i} \le x_{[n+1],i}$.

(2) Consider thresholds within $-\infty$ and all the midpoints $\frac{x_{[n],i}+x_{[n+1],i}}{2}$. Test those thresholds with $s \in \{-1, +1\}$ to determine the best (s, θ) combination that minimizes E^{u}_{in} using feature i.

(3) Pick the best (s, i, θ) combination by enumerating over all possible i.

For those interested in algorithms (who isn't? :-)), step 2 can be carried out in $O(N)$ time only!!

17. (*) What is the value of $\min_{1 \le 1000 \le t} E_{\text{in}}(g_t)$? Note that we are talking about E_{in}, not $E^{\mathbf{u}}_{\text{in}}$. Choose the closest answer; provide your code.

- [a] 0.10
- [b] 0.20
- [c] 0.30
- [d] 0.40
- [e] 0.50

18. (*) What is the value of $\max_{1\leq 1000\leq t} E_{\text{in}}(g_t)$? Note that we are talking about E_{in}, not $E_{\text{in}}^{\mathbf{u}}$. Choose the closest answer; provide your code.

[a] 0.190

[b] 0.380

[c] 0.570

[d] 0.760

[e] 0.950

19. (*) What is the value of $E_{\text{in}}(G)$? Choose the closest answer; provide your code.

[a] 0.00

[b] 0.15

[c] 0.30

[d] 0.45

[e] 0.60

20. (*) What is the value of $E_{\text{out}}(G)$? Choose the closest answer; provide your code.

[a] 0.0028

[b] 0.0056

[c] 0.0084

[d] 0.0112

[e] 0.0140