OSU CSE 3521/5521
Homework #2: Problem Set

Release Date: September 25th, 2020

Submission Instructions

Due Date: October 12st (23:59 ET), 2020

Submission: Please submit your solutions in a single PDF file named HW_2_name.number.pdf (e.g., HW_2_chao.209.pdf) to Carmen. You may write your solutions on paper and scan it, or directly type your solutions and save them as a PDF file. *Submission in any other format will not be graded.*

We highly recommend that you write down the derivation of your answers, and highlight your answers clearly!

Collaboration: You may discuss with your classmates. However, you need to write your own solutions and submit them separately. Also in your written report, you need to list with whom you have discussed for each problem. Please consult the syllabus for what is and is not acceptable collaboration.

Calculation: Please perform rounding to your results after the second decimal number. For example, 1.245 becomes 1.25 and -1.228 becomes -1.23.

1 Adversarial Search [35 points]

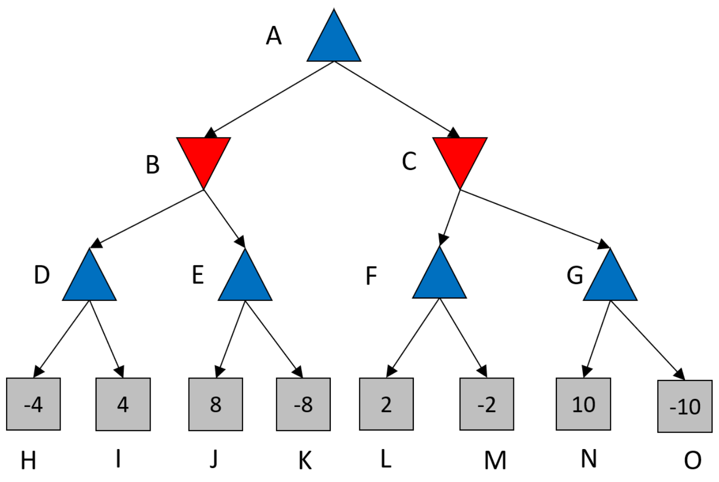

Figure 1: An adversarial search tree

Figure 1 shows an adversarial search tree, in which the blue nodes (i.e., “A”, “D”, “E”, “F”, “G”) are max nodes and the red nodes (i.e., “B”, “C”) are min nodes. The gray nodes are the goal states, each has its utility value (the *higher*, the better). Each max or min node has two actions, going left or right. For example, if you are at “A”, then you can go left to arrive at “B” or go right to arrive at “C”.

Note that, you are provided with this entire adversarial search tree, which has not been traversed/expanded yet, and your job is to do minimax search, determining each node’s value and finding out the minimax solution from “A” to the goal states.

1. Please perform the **adversarial (minimax) search** to

   (a) fill in the value of each node; [**7 points**]

A	B	C	D	E	F	G

   (b) find out the minimax predicted state-sequence. A predicted state-sequence is a ”path” from “A” to the gray (leaf) nodes. For example, “A”-“C”-“G”-“N” (or equivalently, right-right-left) can be a valid predicted state-sequence BUT it may not be the minimax predicted state-sequence. [**3 points**]

2. Please apply the α-β **pruning** to redo the minimax search. Note that, in α-β pruning, you are expanding/traversing the tree in a depth-first-search fashion, in which you will always choose the left action first and then the right action. In other words, the node expansion order will be "A"-"B"-"D"-"H"-"I"-"E"-"J"-"K"-"C"-"F"-"L"-"M"-"G"-"N"-"O", but some nodes might be pruned/skipped without being visited. Every min or max node will have its (α, β), which are initialized by its parent and may be updated when its children are being expanded. The initial (α, β) to the node "A" is $(-\infty, \infty)$.

   Please see the lecture slides for more details. Specifically, please follow the "Alpha-Beta Implementation" slide to update each node's value and its (α, β). The slide provides an recursive implementation for the depth-first-search fashion. Note that, (α, β) in that slide are called/passed by values.

   (a) Fill in the value of each node. If a node is pruned (i.e., no need to visit according to α-β pruning), please put X as the answer for that node. For a gray node, if it is visited, simply fill in its utility value. Note that, the values of the nodes can be ∞ or $-\infty$ and can be different from the previous question. [**15 points**]

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

   (b) Find out the minimax predicted state-sequence. A predicted state-sequence is a "path" from "A" to the gray (leaf) nodes. For example, "A"-"C"-"G"-"N" (or equivalently, right-right-left) can be a valid predicted state-sequence BUT it may not be the minimax predicted state-sequence. [**3 points**]

3. Now let us consider the **expectimax search** by changing each red min node to an expectation node. The probability for the left and right actions are 50% and 50%. Please fill in the value of each node. [**7 points**]

A	B	C	D	E	F	G

2 Bag of words (BoW) and distances [13 points]

1. **[6 points] BoW construction**: Given a dictionary { "capital": 1, "state": 2, "team": 3, "basketball": 4, "hokey": 5, "professional": 6, "bank": 7}, derive the BoW representation for the following three sentences (without normalization):

   A: ["sacramento" "is" "a" "state" "capital" "and" "it" "has" "a" "professional" "basketball" "team"],

   B: ["Columbus" "is" "a" "state" "capital" "and" "it" "has" "a" "professional" "hokey" "team" "but" "no" "professional" "basketball" "team"],

   C: ["the" "capital" "bank" "at" "Cincinnati" "has" "a" "professional" "team"]

   Note that, if a word shows up multiple times in the sentence, you should count it multiple times in the BoW representation. In other words, the representation can contain real numbers (not necessarily binary).

   You MUST get this question right before moving on to 2.2 and 2.3. If your answers for 2.1 are wrong, your answers for 2.2 and 2.3 will be automatically ZERO as their answers are built upon 2.1.

2. **[2 points]** L_1 **distance**: Compute the L_1 distance between A and B, and A and C.

3. **[5 points]** L_1 **normalization**: Perform L_1 normalization to each BoW representation, and compute the L_1 distance between A and B, and A and C.

4. **[0 points]** Do you see some disadvantages of vanilla Bow representations? For example, without 2-gram or higher-gram, the representation cannot tell the different meanings of "capital". With some tokens ignored in the dictionary (like "no"), we cannot tell if Columbus has a basketball team or not. Even with "no" in the dictionary, we cannot simply tell if Columbus does not have a hokey team or a basketball team.

3 Histogram and Parzen window [20 points]

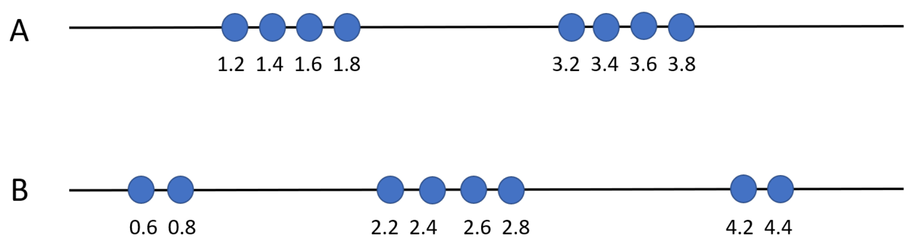

Figure 2: Two datasets, A and B. Each data instance has one feature variable.

Figure 2 shows two datasets, each with 8 data instances.

1. **[5 points]** Given intervals $[0, 1)$, $[1, 2)$, $[2, 3)$, $[3, 4)$, and $[4, 5)$, please construct the L_1-normalized histograms for A and B. You can write each histogram as a 5-dimensional vector, where the first/second/third/fourth/fifth element corresponds to interval $[0, 1)/[1, 2)/[2, 3)/[3, 4)/[4, 5)$. Compute the L_1 distance between the L_1-normalized histograms of A and B.

2. **[5 points]** Given intervals $[0.5, 1.5)$, $[1.5, 2.5)$, $[2.5, 3.5)$, and $[3.5, 4.5)$, please construct the L_1-normalized histograms for A and B. You can write each histogram as a 4-dimensional vector, where the first/second/third/ fourth element corresponds to $[0.5, 1.5)/[1.5, 2.5)/[2.5, 3.5)/[3.5, 4.5)$. Compute the L_1 distance between the L_1-normalized histograms of A and B.

3. **[5 points]** Given intervals $[0.5, 1)$, $[1, 1.5)$, $[1.5, 2)$, $[2, 2.5)$, $[2.5, 3)$, $[3, 3.5)$, $[3.5, 4)$, and $[4, 4.5)$, please construct the L_1-normalized histograms for A and B. You can write each histogram as a 8-dimensional vector, following the order of the intervals. Compute the L_1 distance between the L_1-normalized histograms of A and B.

4. **[0 points]** You will see that the histogram representation is sensitive to the bin (interval) locations and sizes.

5. [**5 points**] Now for dataset A, consider a kernel (see Figure 3)

$$k(u') = \begin{cases} 2 + 4 \times u', \text{ if } -0.5 \leq u' \leq 0; \\ 2 - 4 \times u', \text{ if } 0 < u' \leq 0.5; \\ 0, \text{ otherwise,} \end{cases}$$

please compute the probability density $p(u)$ to see a value u in the future:

(a) $u = 1.5$;

(b) $u = 2.5$.

The definition of $p(u)$ is $\frac{1}{N} \sum_{n=1}^{N} k(u - u_n)$, where N is the total number of training data instances and u_n is the n-th data instance.

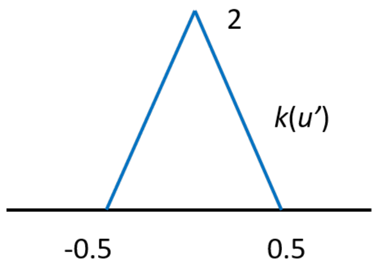

Figure 3: A kernel with a triangular shape.

6. [**0 points**] You will see that to perform kernel density estimation for a value u, you have to keep all the training data.

4 Covariance, z-score, and whitening [16 points]

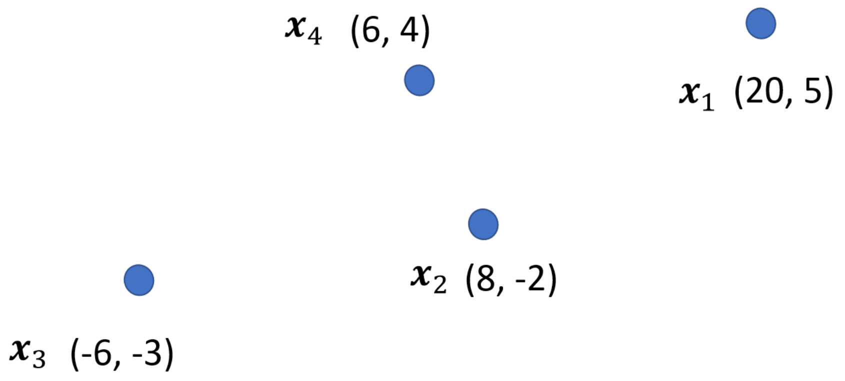

Figure 4: A dataset with four data instances. Each data instance is with two dimensions.

Figure 4 gives a dataset of two dimensions: $\{\boldsymbol{x}_1 = [20, 5]^\top, \boldsymbol{x}_2 = [8, -2]^\top, \boldsymbol{x}_3 = [-6, -3]^\top, \boldsymbol{x}_4 = [6, 4]^\top\}$.

1. [**4 points**] Please compute the covariance matrix $\boldsymbol{\Sigma}$ of Figure 4. $\Sigma[d, d'] = \frac{1}{N}\sum_n (x_n[d] - \mu[d])(x_n[d'] - \mu[d'])$, where $\boldsymbol{\mu}$ is the mean vector. Please make sure you divide by N (i.e., the number of data instances, 4) rather than $N - 1$.

2. [**4 points**] Please compute the vectors $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_4\}$ after applying Z-score transformation to the dataset in Figure 4. Denote $\boldsymbol{\mu}$ the mean vector of the four data points and σ_d the standard deviation of the d-th dimension, the formula of Z-score is

$$\boldsymbol{z}_n[d] = (\boldsymbol{x}_n[d] - \boldsymbol{\mu}[d])/\sigma_d.$$

Please make sure you divide by N (i.e., the number of data instances, 4) rather than $N - 1$ in computing the standard deviation σ_d. That is, you should use the population standard deviation as shown in the slides, not the sample standard deviation.

3. [**4 points**] What is the mean vector and what is the standard deviation of each dimension of the resulting $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_4\}$?

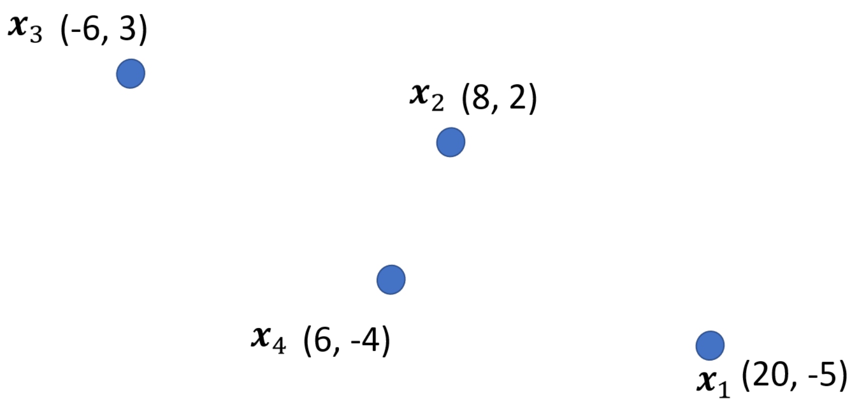

Figure 5: A dataset with four data instances. Each data instance is with two dimensions.

Figure 5 gives a dataset of two dimensions: $\{\boldsymbol{x}_1 = [20, -5]^\top, \boldsymbol{x}_2 = [8, 2]^\top, \boldsymbol{x}_3 = [-6, 3]^\top, \boldsymbol{x}_4 = [6, -4]^\top\}$.

The mean vector $\boldsymbol{\mu}$ is $[7, -1]^\top$. The covariance $\boldsymbol{\Sigma}$ is $\begin{bmatrix} 85 & -24.5 \\ -24.5 & 12.5 \end{bmatrix}$. $\boldsymbol{\Sigma}^{-0.5} = \begin{bmatrix} 0.133 & 0.096 \\ 0.096 & 0.418 \end{bmatrix}$.

[**4 points**] Please applying whitening to the dataset in Figure 5. The formula of whitening is

$$\boldsymbol{z}_n = \boldsymbol{\Sigma}^{-0.5}(\boldsymbol{x}_n - \boldsymbol{\mu}).$$

What are the resulting vectors $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_4\}$?

Note: The dataset used for whitening is different from the one used for 4.1, 4.2, and 4.3.
Note: For 4.2 and 4.4, each $\boldsymbol{z}_n$ is 1 point and the entire vector must be correct to get that 1 point.

5 Linear regression [16 points]

Recall that in the lecture, we show that the solution of linear regression has a closed form solution $\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top)^{-1}\tilde{\boldsymbol{X}}\boldsymbol{y}$, where $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_N]$, $\tilde{\boldsymbol{x}}_n = [\boldsymbol{x}_n^\top, 1]^\top$, and $\boldsymbol{y} = [y_1, \cdots, y_N]^\top$. One potential problem is that $\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top$ may not be invertible.

Now let us consider a different minimization problem to find $\tilde{\boldsymbol{w}}$. Again, let $E(\tilde{\boldsymbol{w}}) = \sum_n (\tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{x}}_n - y_n)^2$. Instead of minimize $E(\tilde{\boldsymbol{w}})$, we are going to minimize $E(\tilde{\boldsymbol{w}}) + \lambda \times \|\tilde{\boldsymbol{w}}\|_2^2 = E(\tilde{\boldsymbol{w}}) + \lambda \times \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{w}}$, where $\lambda > 0$.

[16 points] Please show that the solution $\tilde{\boldsymbol{w}}$ now is $\tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top + \lambda \boldsymbol{I})^{-1}\tilde{\boldsymbol{X}}\boldsymbol{y}$. $\boldsymbol{I}$ is an $(D+1) - \text{by} - (D+1)$ identity matrix, where D is the dimensionality of $\boldsymbol{x}_n$. (no more than 8 lines of derivation)

In machine learning, this solution has a name rigid regression. Note that, now $\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top + \lambda \boldsymbol{I}$ is invertible.