OSU CSE 3521
Homework #2: Problem Set

Release Date: September 21st, 2021

Submission Instructions

Due Date: October 12th (23:59 ET), 2021

Submission: Please submit your solutions in a single PDF file named HW2_name_number.pdf (e.g., HW2_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 at a very high level. However, you need to write your own solutions and submit them separately by yourself. Also in your written report, you need to list with whom you have discussed for each problem (please do so in the first page). Please consult the syllabus for what is and is not an acceptable collaboration.

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

1 Bag of words (BoW) and distances [7 points]

1. **[3 points] BoW construction**: Given a dictionary { "capital": 1, "state": 2, "team": 3, "basketball": 4, "hokey": 5, "professional": 6, "bank": 7}, derive the BoW representation (i.e., a 7-dimensional vector) 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 answer this question correctly before moving on to 1.2 and 1.3, since their answers are built upon 1.1. If your answers for 1.1 are wrong, your will not earn any point for your answers for 1.2 and 1.3.

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

3. **[2 points]** L_1 **normalization**: Perform L_1 normalization to the BoW representations of A, B, and C first, and compute the L_1 distance between A and B, and between 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.

2 Histogram and Parzen window [10 points]

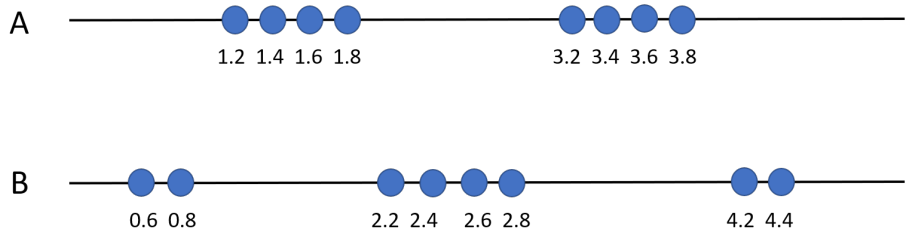

Figure 1: Two datasets, A and B. Each data instance has one feature variable.

Figure 1 shows two datasets, each with 8 data instances.

1. **[2 points]** Given intervals $[0, 1)$, $[1, 2)$, $[2, 3)$, $[3, 4)$, and $[4, 5)$, please first 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 the interval $[0, 1)/[1, 2)/[2, 3)/[3, 4)/[4, 5)$. Then, compute the L_1 distance between the L_1-normalized histograms of A and B.

2. **[2 points]** Given intervals $[0.5, 1.5)$, $[1.5, 2.5)$, $[2.5, 3.5)$, and $[3.5, 4.5)$, please first 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 $[0.5, 1.5)/[1.5, 2.5)/[2.5, 3.5)/[3.5, 4.5)$. Then, compute the L_1 distance between the L_1-normalized histograms of A and B.

3. **[2 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 first construct the L_1-normalized histograms for A and B. You can write each histogram as a 8-dimensional vector, following the ascending order of the intervals. Then, compute the L_1 distance between the L_1-normalized histograms of A and B.

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

5. **[4 points]** Now for dataset A, consider a kernel fuction (see Figure 2)

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

please compute the probability density $p(u)$ for seeing 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 in A and u_n is the n-th data instance.

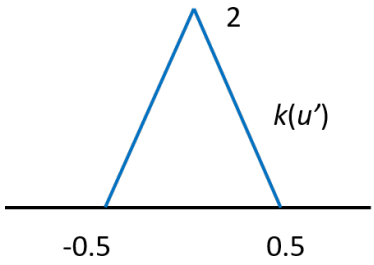

Figure 2: A kernel with a triangular shape.

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

3 Covariance, z-score, whitening, and PCA [10 points]

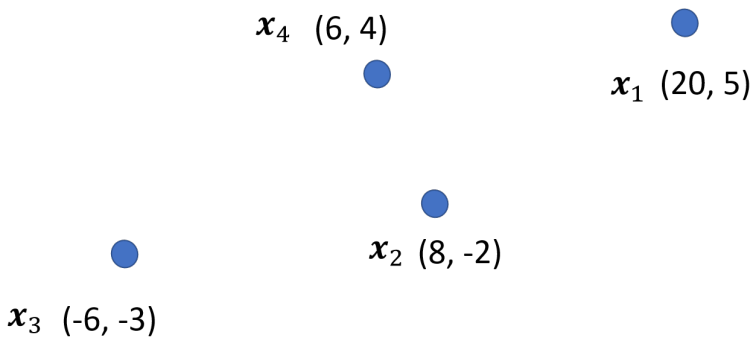

Figure 3: A dataset with four data instances. Each data instance is 2-dimensional.

Figure 3 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. **[2 points]** Please compute the covariance matrix $\boldsymbol{C}$ of Figure 3. Here, we use $\boldsymbol{C}$ to denote a covariance matrix to prevent any confusion. $C[d, d'] = \frac{1}{N}\sum_n (x_n[d] - \mu[d])(x_n[d'] - \mu[d'])$, where $\boldsymbol{\mu}$ is the 2-dimensional mean vector of the four instances. Please make sure you divide by N (i.e., the number of data instances, 4) rather than $N - 1$.

2. **[2 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 3. Denote by $\boldsymbol{\mu}$ the 2-dimensional mean vector of the four instances and by 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. **[2 points]** What is the 2-dimensional mean vector and what is the standard deviation of each dimension of the resulting $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_4\}$?

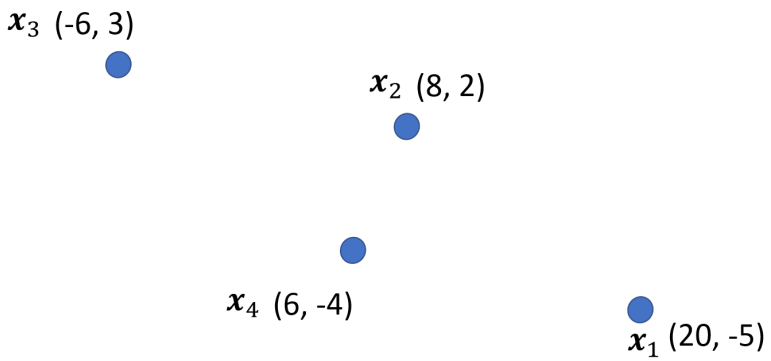

Figure 4: A dataset with four data instances. Each data instance is 2-dimensional.

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

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

4. **[2 points]** Please applying whitening to the dataset in Figure 4. The formula of whitening is

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

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

5. **[2 points]** Please apply PCA to the dataset in Figure 4 without reducing the dimensionality. That is, you are to construct the 2-by-2 matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2]$, where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are two 2-dimensional eigenvectors ($\boldsymbol{w}_1$ has the larger eigenvalue; $\boldsymbol{w}_1$ has the smallest eigenvalue). Please L_2-normalize each vector. You may use python or other software to compute $\boldsymbol{W}$.

4 Linear regression [8 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 $\lambda > 0$.

[8 points] Please show that the solution $\tilde{\boldsymbol{w}}$ which minimizes $E(\tilde{\boldsymbol{w}}) + \lambda \times \tilde{\boldsymbol{w}}^\top \tilde{\boldsymbol{w}}$ 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 $\boldsymbol{x}_n$. Please write down your derivation with no more than 8 lines.

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.

5 Regression, Gauss-Newton, and gradient descent: Part-1 [12 points]

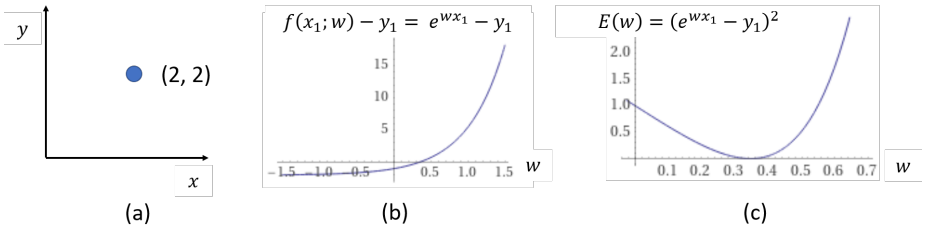

Figure 5: (a) A dataset of one data instance; (b) the corresponding $f(x_1; w) - y_1$; (c) the corresponding $E(w) = (f(x_1; w) - y_1)^2$.

Figure 5 (a) shows a dataset of just one data instance: $(x_1, y_1) = (2, 2)$. Now we want to fit a nonlinear curve $f(x; w) = e^{wx}$ to the dataset, using the sum of square error (SSE): $E(w) = (f(x_1; w) - y_1)^2$. Figure 5 (b) and (c) show the curve of $f(x_1; w) - y_1$ w.r.t. w and the curve of $E(w)$ w.r.t. w, respectively.

5.1 Gauss-Newton method [6 points in total]

Let us use the Gauss-Newton method introduced in the lectures to find the solution $w^\star$ that minimizes $E(w)$. Note that, the optimal solution is 0.347. Since the Gauss-Newton method is an iterative method, the solution depends on the number of iteration.

1. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform three iterations of Gauss-Newton to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]

2. Begin with the initialization $\hat{w}^{(0)} = 0.0$, perform three iterations of Gauss-Newton to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]

3. Begin with the initialization $\hat{w}^{(0)} = -1.0$, perform three iterations of Gauss-Newton to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]

Your answers must contain numerical values. For example, 0.500 is allowed but 1/2 is not allowed. Please round your answer to the third decimal. For example, 1.333333 becomes 1.333; -1.333333 becomes -1.333.

Do you see that iterative methods are sensitive to the initialization? [0 point]

5.2 Gradient descent [6 points in total]

Let us now apply the gradient descent (GD) method introduced in the lectures to find the solution $w^\star$ that minimizes $E(w)$. Since GD is an iterative method, the solution depends on the number of iteration.

For our problem where w is one-dimensional, given the current guess $\hat{w}^{(t)}$, GD performs the following update:

$$\hat{w}^{(t+1)} \leftarrow \hat{w}^{(t)} - \eta \frac{\mathrm{d}E}{\mathrm{d}w}(\hat{w}^{(t)}), \tag{1}$$

where $\frac{\mathrm{d}E}{\mathrm{d}w}(\hat{w}^{(t)})$ is the derivative computed at $\hat{w}^{(t)}$ and η is the step size or learning rate.

1. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform three iterations of GD (with $\eta = 0.1$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]
2. Begin with the initialization $\hat{w}^{(0)} = 0.0$, perform three iterations of GD (with $\eta = 0.1$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]
3. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform three iterations of GD (with $\eta = 0.001$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]

Your answers must contain numerical values. For example, 0.500 is allowed but 1/2 is not allowed. Please round your answer to the third decimal. For example, 1.333333 becomes 1.333; -1.333333 becomes -1.333.

Do you see that iterative methods are sensitive to the initialization and the step size? [0 point]

6 Regression, Gauss-Newton, and gradient descent: Part-2 [13 points]

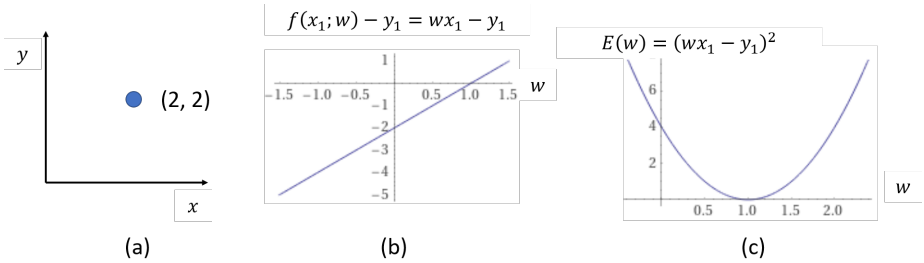

Figure 6: (a) A dataset of one data instance; (b) the corresponding $f(x_1; w) - y_1$; (c) the corresponding $E(w) = (f(x_1; w) - y_1)^2$.

Following Question 5, now let us consider a simpler problem: to fit a linear line $f(x; w) = wx$ to the same data, again using the sum of square error (SSE): $E(w) = (f(x_1; w) - y_1)^2$. Figure 6 (a) shows the same data as Figure 5, but Figure 6 (b) and (c) show the curve of $f(x_1; w) - y_1$ w.r.t. w and the curve of $E(w)$ w.r.t. w, respectively, using $f(x; w) = wx$.

6.1 Gauss-Newton method [6 points in total]

Let us again use the Gauss-Newton method to find the solution $w^\star$ that minimizes $E(w)$ in Figure 6. Note that, the optimal solution is 1.000.

1. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform just **one** iteration of Gauss-Newton to get $\hat{w}^{(1)}$. What is $\hat{w}^{(1)}$ (a numerical value)? [2 point]

2. Begin with the initialization $\hat{w}^{(0)} = 0.0$, perform just **one** iteration of Gauss-Newton to get $\hat{w}^{(1)}$. What is $\hat{w}^{(1)}$ (a numerical value)? [2 point]

Your answers must contain numerical values. For example, 0.500 is allowed but 1/2 is not allowed. Please round your answer to the third decimal. For example, 1.333333 becomes 1.333; -1.333333 becomes -1.333.

Do you see any difference from the observations in Question 5.1? If so, please describe the difference and explain the reason. [2 point]

6.2 Gradient descent [7 points in total]

Let us now apply gradient descent (GD) to find the solution $w^\star$ that minimizes $E(w)$. For our problem where w is one-dimensional, given the current guess $\hat{w}^{(t)}$, GD performs the following update:

$$\hat{w}^{(t+1)} \leftarrow \hat{w}^{(t)} - \eta \frac{\mathrm{d}E}{\mathrm{d}w}(\hat{w}^{(t)}), \tag{2}$$

where $\frac{\mathrm{d}E}{\mathrm{d}w}(\hat{w}^{(t)})$ is the derivative computed at $\hat{w}^{(t)}$ and η is the step size or learning rate.

1. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform three iterations of GD (with $\eta = 0.1$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]
2. Begin with the initialization $\hat{w}^{(0)} = 0.0$, perform three iterations of GD (with $\eta = 0.1$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]
3. Begin with the initialization $\hat{w}^{(0)} = 1.5$, perform three iterations of GD (with $\eta = 1.0$) to get $\hat{w}^{(3)}$. What is $\hat{w}^{(3)}$ (a numerical value)? [2 point]

Your answers must contain numerical values. For example, 0.500 is allowed but 1/2 is not allowed. Please round your answer to the third decimal. For example, 1.333333 becomes 1.333; -1.333333 becomes -1.333.

Can GD obtain the optimal solution in a few iterations? [1 point]