CS 165B – Machine Learning, Spring 2023

Eddie Zhang <ete@ucsb.edu>

Assignment #3

Due Thursday, May 18 by 11:59pm PST

Notes:

- *This assignment is to be done individually. You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.*
- Eddie Zhang <ete@ucsb.edu> is the head TA for this HW, please contact for questions
- Be sure to re-read the "Policy on Academic Integrity" on the course syllabus.
- Be aware of the late policy in the course syllabus – i.e., *late submissions will be accepted but 20% deduction per day.*
- Justify every answer you give – **show the work** that achieves the answer or **explain** your response.
- Any updates or corrections will be posted on Piazza, so check there occasionally.
- **All assignments must be clear and legible.** It is recommended to type the solutions on this Microsoft Word file directly and save it as PDF. If you'll be submitting a handwritten assignment, please ensure that it's readable and neat. If your writing is not easily readable, your solution is not easy to follow on the page, or your PDF is not of very high quality, your assignment will not be graded. DO NOT submit picture of your written work. (If you must scan your written work in, use a high-quality scanner. Plan in advance.)
- To turn in your assignment:
  - Use entry code **2K4D2G** to register as a student for **CS 165B Spring 2023 Machine Learning** on Gradescope.
  - Submit a PDF version to Gradescope

Problem #1 [10 points] Kernel

Suppose the input space is two-dimensional $x = \left[x_1, x_2\right]^T$. Define the kernel function $\kappa(x, z)$ to be $\left(x^T z + 2\right)^2$.

(a) What is the corresponding feature mapping space? Express your answer in vector form in terms of x**.**

(b) Show that this feature mapping space has a dot product which is equivalent to the original kernel function.

(a) the feature mapping $\phi(x) = \left[x_1^2, x_2^2, 2x_1, 2x_2, \sqrt{2}x_1x_2,\ 2\right]^T$

(b)

$$
\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{x}') &\overset{(1)}{=} (2 + \boldsymbol{x} \cdot \boldsymbol{x}')^2 && (2)\\
&= (2 + x_1x_1' + x_2x_2')^2 && (3)\\
&= 4 + 4x_1x_1' + 4x_2x_2' + x_1^2{x_1'}^2 + 2x_1x_1'x_2x_2' + x_2^2{x_2'}^2\,. && (4)
\end{aligned}
$$

The dot product of this feature map

$$(\boldsymbol{z}, \boldsymbol{z}') := z_1z_1' + z_2z_2' + z_3z_3' + z_4z_4' + z_5z_5' + z_6z_6'\,.$$

is equivalent to the above expansion.

Problem #2 [10 points] kNN

Assume that we have two classes, A and B and a new document d to be classified. The following training data is available:

d_i	*class*	$\cos(\vec{v}(d_i), \vec{v}(d))$
d_1	*A*	*1*
d_2	*B*	*0.95*
d_3	*B*	*0.94*
d_4	*A*	*0.45*
d_5	*A*	*0.4*
d_6	*B*	*0.39*

If we're using the cosine as a distance measure (i.e. the higher the cosine, the closer the two vectors), then which class would be assigned to d with a k-nearest neighbor classifier using cosine if we use

1. k = 3 and simple majority vote?
2. k = 5 and simple majority vote?
3. k = 3 and weighted by cosine similarity? This means each vote is weighted by its cosine similarity to d.
4. k = 5 and weighted by cosine similarity?

Solution:
1. k = 3 and simple majority vote: score(A, d) = 1, score(B, d) = 2, therefore class B
2. k = 5 and simple majority vote: score(A, d) = 3, score(B, d) = 2, therefore class A
3. k = 3 and a weighted score as in slide 27: score(A, d) = 1, score(B, d) = 0.95 + 0.94, therefore class B
4. k = 5 and a weighted score as in slide 27: score(A, d) = 1 + 0.45 + 0.4, score(B, d) = 0.95 + 0.94, therefore class B

Problem #3 [10 points] Perceptron

For the following training data in the form = {feature values, label}:

{(-2, 3, 1), -1}
{(4, 1, 2), 1}
{(3, 5, 1), -1}
{(2, 1, 3), 1}
{(6, 2, 1), 1}

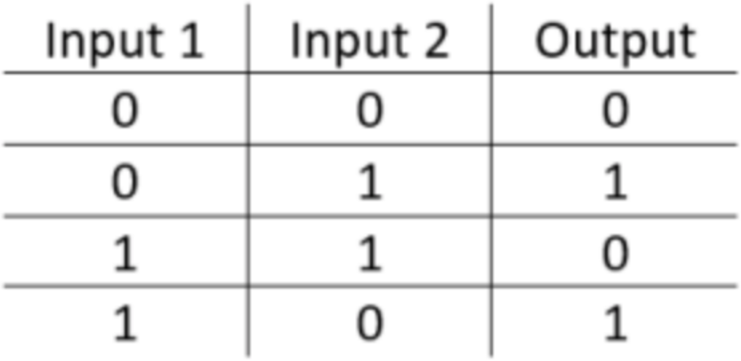
and with an initial weight w = (1, 1, 1),

please show how w will be updated using the Perceptron algorithm with learning rate $\eta = 0.5$. Suppose that there is no bias. You only need to show the results for one iteration on all the data samples, i.e., only need to show five updates on w here.

Step	$y_i W^T x_i$	New *w*
1	-2	(2, -0.5, 0.5)
2	8.5	(2, -0.5, 0.5)
3	-4	(0.5, -3, 0)
4	-2	(1.5, -2.5, 1.5)
5	5.5	(1.5, -2.5, 1.5)

Problem #4 [10 points] Perceptron with XOR data

XOR data is defined as below:



Input 1	Input 2	Output
0	0	0
0	1	1
1	1	0
1	0	1

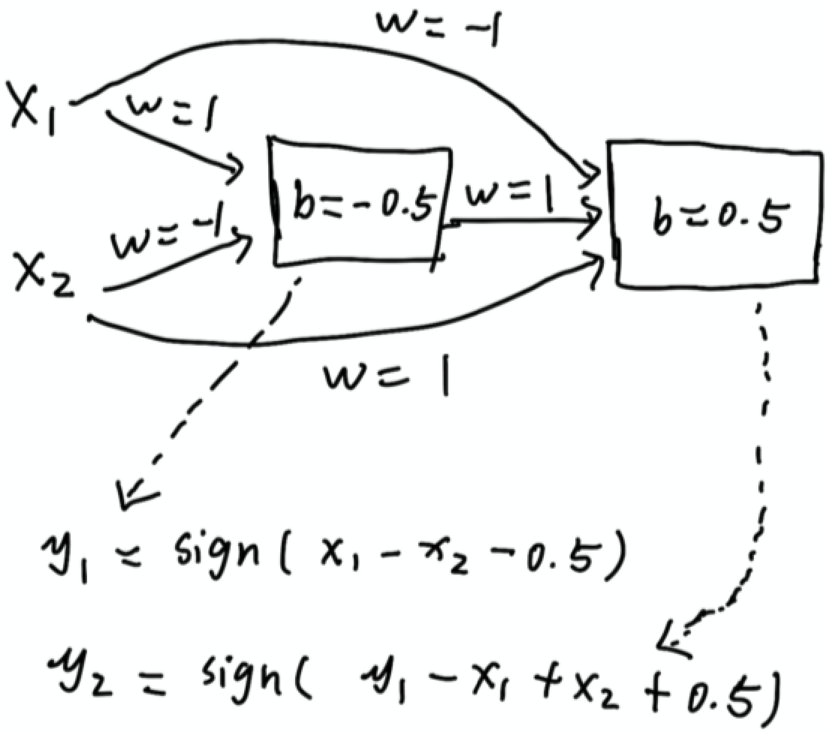
(a) Explain why a Perceptron cannot classify this XOR dataset.

(b) If a Perceptron cannot classify XOR data, is it possible to stack two Perceptrons together so that it can classify XOR data correctly? If yes, draw the network; if not, please explain the reason.

(c) Show a possible solution to handle XOR data (using only one Perceptron)

(a) No. Because it is not linearly separable

(b) Yes. The weight and threshold are plot in the network. Other correct designs are acceptable, but only two perceptrons should be used in the network.

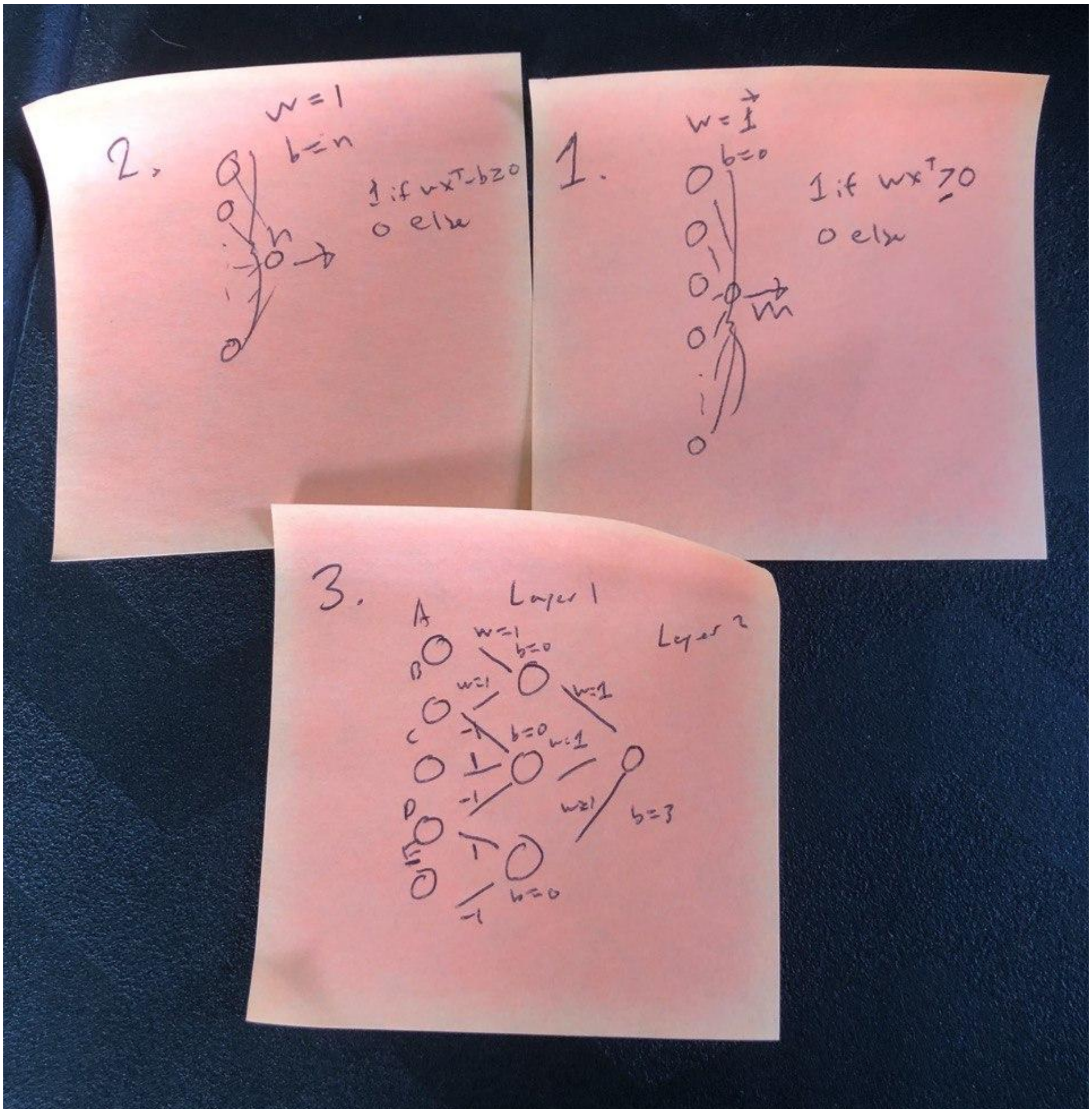

$w = -1$
X_1
$w = 1$
$b = -0.5$
$w = 1$
$b = 0.5$
$w = -1$
X_2
$w = 1$
$y_1 = \text{sign}(x_1 - x_2 - 0.5)$
$y_2 = \text{sign}(y_1 - x_1 + x_2 + 0.5)$

(c) Use Kernel function to map the input space in a feature space. For example k(x1, x2) = x1*x2 + (x1-1)*(x2-1)

Problem #5 [8 points] Perceptron

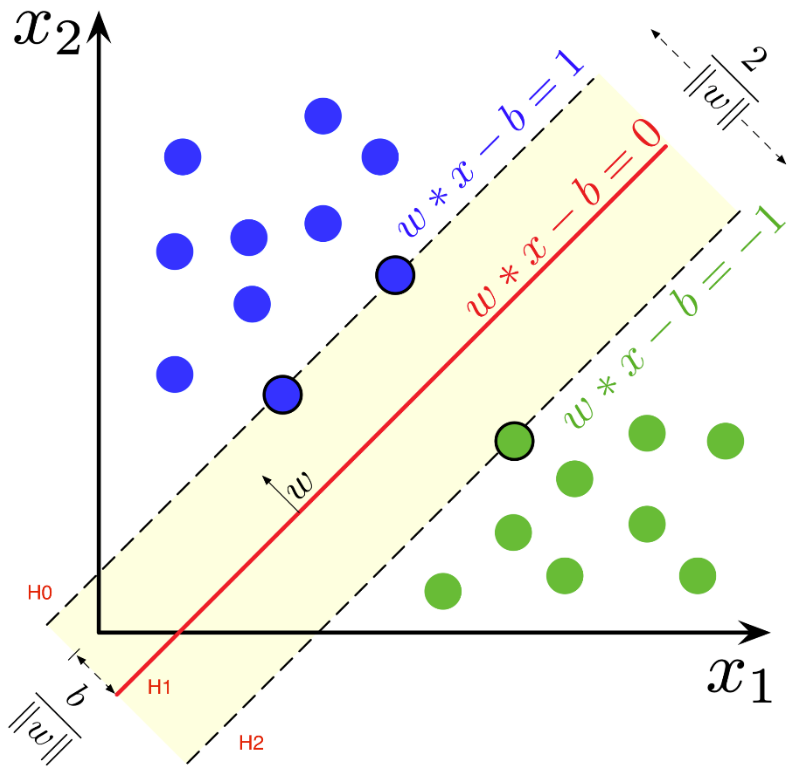
Let's assign False=0 and True=1. Please show how each of the following could be constructed with a Perceptron:

- Compute the OR function of m inputs
- Compute the AND function of n inputs
- 2-stacked perceptrons for computing the function $(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$
- 2-stacked perceptrons to compute any (given) logical expression, assuming it is written in Conjunctive Normal Form.



4. same as 3, just use 1 for true, -1 for false in first layer and then use w = 1 for all the neurons in second layer.

Problem #6 [10 points] SVM



In real world applications, there are outliers in data. This can be dealt using a soft margin, specified in a slightly different optimization problem as below (soft-margin SVM):

$$\min_w \frac{1}{2}(w^T w) + C\sum_i^N \xi_i \quad where\ \xi_i \geq 0 \quad s.t. \quad y^{(i)}(w^T x^{(i)} - b) \geq 1 - \xi_i$$

ξ_i represents the slack for each data point i, which allows misclassification of data points in the event that the data is not linearly separable. Note that (H0, H1, H2) are the hyperplanes as shown in the above graph. Let us now consider a positively labeled datapoint (i.e., $y^{(i)} = 1$).

(a) Intuitively, where could this data point lie relative to (H0 , H1, H2) when $\xi_i = 0$? Is this data point classified correctly?

(b) Where could this data point lie relative to (H0 , H1, H2) when $0 < \xi_i < 1$? Is this data point classified correctly?

(c) Where does this data point lie relative to (H0 , H1, H2) when $\xi_i = 1$?

(d) Finally, where does it lie relative to (H0 , H1, H2) when $\xi_i > 1$? Is this data point classified correctly?

(a) Left side of or on H0. Correctly

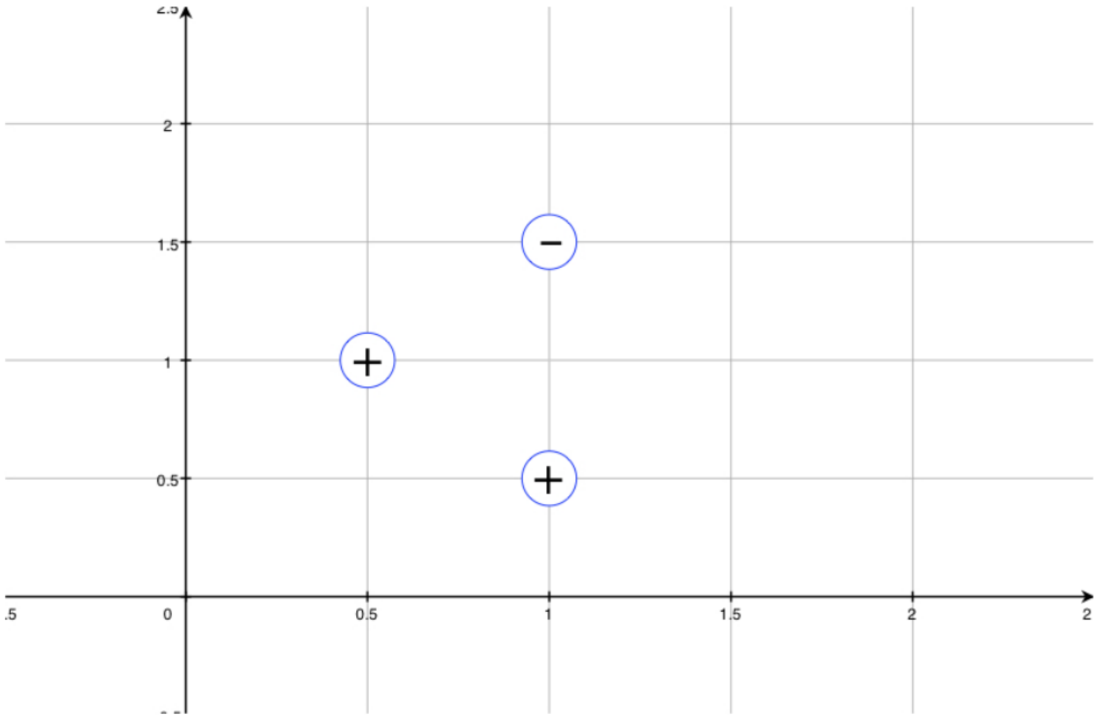
(b) Between H0 and H1. Correctly

(c) On H1

(d) On the right side of H1. Not correctly.

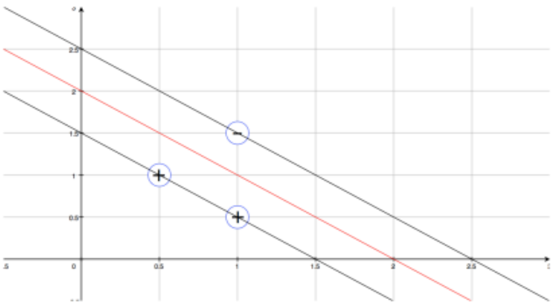
Problem #7 [6 points] SVM

Take a look at this image that presents three input vectors in two dimensions with linear separability. Find the optimal linear support vector machine weights and equation that separates the categories by maximizing the margin. Also write the equation for the H_+ and H_- hyperplanes.



Solution

All three data points are support vectors. The margin hyperplan H_+ is the line passing through the two positive points. The margin hyperplan H_- is the line passing through the negative point that is parallel to H_+. The decision boundary is the red line "half way" between H_+ and H_-. The equation of the decision boundary is –x+2=0. The following picture illustrates the solution:



Problem #8 [Optional 10 points] SVM

(a) Define a linear classifier $f(x) = w^T x - t,$ where x is classified into class -1 if $f(x) < 0$ and class +1 otherwise. We define the margin of x as in the lecture notes: $Margin(x) = \frac{m}{||w||} = min_i\,[y_i f(x_i)]\frac{1}{||w||}$. Assume there are n samples, SVM tries to maximize the margin. Therefore, we have the following objective function w.r.t w:

$$\max_w \frac{min_i\,[y_i f(x_i)]}{||w||_2} \qquad (1)$$

Show that equation (1) is equivalent to the following problem:

$$\min \frac{1}{2}||w||_2^2$$
$$subject\ to\ y_i\left(w^T x_i - t\right) \geq 1,\ \text{i=1,…,n}$$

(b) For soft-margin SVM, we introduce a slack variable and a parameter C in the optimization problem:

$$\min \frac{1}{2} w^T w + C \sum \xi_i \quad s.t.\ \xi_i \geq 0,\ \ y_i(w^T x_i) \geq 1 - \xi_i$$

Explain how C affects the margin and the margin error.

(a) Substitute $y_i f_i(x_i)$ with m and the inner minimization with inequality $y_i f_i(x_i) \geq m$

The max problem becomes:

$$max \frac{m}{\|w\|_2}, \ subject\ to\ y_i f_i(x_i) \geq m$$

Then we flip the max to min by flipping the objective function (OK since it is monotonically decreasing). Multiplying by ½ doesn't change the sign so we do this as well.

$$\min \frac{1}{2}\|w\|_2^2$$
$$subject\ to\ y_i f_i(x_i) \geq m$$

Finally, we can set m = 1 since m, t, and ||w|| are scale-invariant w.r.t each other.

(b) If C is large, the optimization problem strengthens on minimizing the slack variable, and thus give less tolerance to misclassified samples. Thus the margin and margin error becomes smaller. If C is smaller, SVM allows more samples to be misclassified and thus having larger margin and margin error. The extreme case is $C = 0$ and our soft-margin SVM becomes hard-margin SVM.