Define $\mathbb{R}^2$ as the set of 2D points. We will represent a point (x, y) as a 2D vector. Parameterized by a 2D vector $\boldsymbol{w}$, a linear classifier h maps a point $\boldsymbol{p}$ to 1 if $\boldsymbol{w} \cdot \boldsymbol{p} \geq 0$, or -1 otherwise.

Problem 1. In the seminar, we introduced the Perceptron algorithm for *online* learning. Suppose that the algorithm initially holds a linear classifier (i.e., h_{now}) parameterized by $\boldsymbol{w} = (0, 0)$. Use the algorithm to process (in the online model) the following sequence of points:

point $\boldsymbol{a} = (1, 2)$, label -1
point $\boldsymbol{b} = (-2, 3)$, label -1
point $\boldsymbol{c} = (2, 4)$, label 1

Given the value of $\boldsymbol{w}$ after processing each point.

Solution.

- Processing $\boldsymbol{a}$: Since $\boldsymbol{w} \cdot \boldsymbol{a} = 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(0, 0) - (1, 2) = (-1, -2)$.
- Processing $\boldsymbol{b}$: Since $\boldsymbol{w} \cdot \boldsymbol{b} < 0$, the point is correctly classified. Hence, $\boldsymbol{w}$ incurs no changes.
- Processing $\boldsymbol{c}$: Since $\boldsymbol{w} \cdot \boldsymbol{c} < 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(-1, -2) + (2, 4) = (1, 2)$.

Problem 2. What is the worst-permutation mistake bound of the Perceptron algorithm on the point set $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$? Here, $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are the same as given in Problem 1. You should assume that the algorithm *always* starts with $\boldsymbol{w} = (0, 0)$.

Solution. Consider how Perceptron processes the permutation $\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{b}$:

- Processing $\boldsymbol{a}$: Since $\boldsymbol{w} \cdot \boldsymbol{a} = 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(0, 0) - (1, 2) = (-1, -2)$.
- Processing $\boldsymbol{c}$: Since $\boldsymbol{w} \cdot \boldsymbol{c} < 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(-1, -2) + (2, 4) = (1, 2)$.
- Processing $\boldsymbol{b}$: Since $\boldsymbol{w} \cdot \boldsymbol{b} > 0$, the point is mis-classified.

We can now conclude that the worst-permutation mistake bound is 3.

Problem 3. We learned a method to convert Perceptron to a batch learning algorithm. Let us run the converted Perceptron algorithm on the set $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$. Recall that the method runs in iterations. Assume that, in each iteration, the points remaining in the set are processed in alphabetic order. What is the final $\boldsymbol{w}$ output by the algorithm?

Solution. Iteration 1:

- Processing $\boldsymbol{a}$: Since $\boldsymbol{w} \cdot \boldsymbol{a} = 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(0, 0) - (1, 2) = (-1, -2)$. The point is deleted.
- Processing $\boldsymbol{b}$: Since $\boldsymbol{w} \cdot \boldsymbol{b} < 0$, the point is correctly classified. Hence, $\boldsymbol{w}$ incurs no changes.
- Processing $\boldsymbol{c}$: Since $\boldsymbol{w} \cdot \boldsymbol{c} < 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(-1, -2) + (2, 4) = (1, 2)$. The point is deleted.

Now the set contains only $\boldsymbol{b}$. Iteration 2 proceeds as follows:

- Processing $\boldsymbol{b}$: Since $\boldsymbol{w} \cdot \boldsymbol{b} > 0$, the point is mis-classified. Hence, $\boldsymbol{w}$ changes to $(1, 2) - (-2, 3) = (3, -1)$.

This is the final $\boldsymbol{w}$ returned.