Problem 1. Let $P$ be a set of 4 points: $A = (1, 2), \ B = (2, 1), \ C = (0, 1) \text{ and } D = (1, 0)$ where $A, B$ have color red, while $C, D$ have color blue. We want to find a plane that separates the red points from the blue points.

1. Convert the problem to $\mathbb{R}^3$ so that it can be solved by the Perceptron algorithm. Give the resulting dataset $P'$.

2. Execute Perceptron on $P'$. Give the equation of the plane that is maintained by the algorithm at the end of each iteration.

3. Convert the plane output by Perceptron back to the original 2d space to obtain a separation plane on $P$.

Answer.

1. $P'$ includes the following points: $A' = (1, 2, 1), \ B' = (2, 1, 1), \ C' = (0, 1, 1), \ D' = (1, 0, 1)$.

2. Let us represent the plane maintained by Perceptron as $c_1 x_1 + c_2 x_2 + c_3 x_3 = 0$. Denote by $\vec{c} = (c_1, c_2, c_3)$. At the beginning of Perceptron, $\vec{c} = (0, 0, 0)$.

   We use $\vec{A}'$ to denote the vector $(1, 2, 1)$, obtained by listing the coordinates of $A'$. Define $\vec{B}', \vec{C}', \vec{D}'$ similarly.

   **Iteration 1.** Since $\vec{A}'$ does not satisfy $\vec{A}' \cdot \vec{c} > 0$, we update $\vec{c}$ to $\vec{c} + \vec{A}' = (0, 0, 0) + (1, 2, 1) = (1, 2, 1)$.

   **Iteration 2.** Since $\vec{C}'$ does not satisfy $\vec{C}' \cdot \vec{c} < 0$, we update $\vec{c}$ to $\vec{c} + \vec{C}' = (1, 2, 1) - (0, 1, 1) = (1, 0, 1)$.

   **Iteration 3.** Since $\vec{C}'$ does not satisfy $\vec{C}' \cdot \vec{c} < 0$, we update $\vec{c}$ to $\vec{c} - \vec{C}' = (1, 1, 0) - (0, 1, 1) = (1, 0, -1)$.

   **Iteration 4.** Since $\vec{A}'$ does not satisfy $\vec{A}' \cdot \vec{c} > 0$, we update $\vec{c}$ to $\vec{c} + \vec{A}' = (1, 0, -1) + (1, 2, 1) = (2, 2, 0)$.

   **Iteration 5.** Since $\vec{C}'$ does not satisfy $\vec{C}' \cdot \vec{c} < 0$, we update $\vec{c}$ to $\vec{c} - \vec{C}' = (2, 2, 0) - (0, 1, 1) = (2, 1, -1)$.

   **Iteration 6.** Since $\vec{C}'$ does not satisfy $\vec{C}' \cdot \vec{c} < 0$, we update $\vec{c}$ to $\vec{c} - \vec{C}' = (2, 1, -1) - (0, 1, 1) = (2, 0, -2)$.

   **Iteration 7.** Since $\vec{A}'$ does not satisfy $\vec{A}' \cdot \vec{c} > 0$, we update $\vec{c}$ to $\vec{c} + \vec{A}' = (2, 0, -2) + (1, 2, 1) = (3, 2, -1)$.

   **Iteration 8.** Since $\vec{C}'$ does not satisfy $\vec{C}' \cdot \vec{c} < 0$, we update $\vec{c}$ to $\vec{c} - \vec{C}' = (3, 2, -1) - (0, 1, 1) = (3, 1, -2)$. 


**Iteration 9.** Since \(\vec{D}'\) does not satisfy \(\vec{D}' \cdot \vec{c} < 0\), we update \(\vec{c}\) to \(\vec{c} - \vec{D}' = (3,1,-2)-(1,0,1) = (2,1,-3)\).

**Iteration 10.** No more violation. So we have found a separation plane \(2x_1 + x_2 - 3x_3 = 0\) for \(P'\).

3. The corresponding separation plane in the original 2d space is therefore \(2x_1 + x_2 - 3 = 0\).

**Problem 2.** Let \(P\) be a set of multidimensional points where each point is colored in either red or blue. We want to design an algorithm to achieve the following purpose:

- Either return a separation plane;
- Or declare that \(P\) has no separation planes with a margin at least \(\gamma\) (recall that the margin of a separation plane \(\pi\) equals the minimum of the distances from the points of \(P\) to \(\pi\)).

Note that your algorithm must still work even if \(P\) is not linearly separable.

**Answer.** Simply run Perceptron, and return whatever plane found by the algorithm. If the algorithm still has not finished after \(R^2/\gamma^2\) vector corrections, manually force it to stop, and declare that no separation plane has a margin at least \(\gamma\).

**Problem 3.** Describe how to solve the classification problem in Problem 1 by way of linear programming.

**Answer.**
First convert the problem into \(\mathbb{R}^3\) by creating a dataset \(P'\) which contains the following points: \(A' = (1,2,1), B' = (2,1,1), C' = (0,1,1), D' = (1,0,1)\). In particular, \(A'\) and \(B'\) are red, whereas \(C'\) and \(D'\) are blue.

Suppose that there exists a separation plane \(c_1x_1 + c_2x_2 + c_3x_3 = 0\). If \((x_1,x_2,x_3)\) is a red point in \(P'\), we require that it should satisfy \(c_1x_1 + c_2x_2 + c_3x_3 \geq c_4\). If \((x_1,x_2,x_3)\) is a blue point in \(P'\), we require it should satisfy \(c_1x_1 + c_2x_2 + c_3x_3 \leq -c_4\). This gives us four inequalities:

\[
\begin{align*}
  c_1 + 2c_2 + c_3 & \geq c_4 \\
  2c_1 + c_2 + c_3 & \geq c_4 \\
  c_2 + c_3 & \leq -c_4 \\
  c_1 + c_3 & \leq -c_4
\end{align*}
\]

We want to find \(c_1, c_2, c_3, c_4\) such that

- all the above inequalities are satisfied
- \(c_4\) is maximized.

This is an instance of LP (with the objective function to maximize \(c_4\)). We check whether the \(c_4\) returned by LP is strictly greater than 0. If so, we conclude that the original \(P\) is not linearly separable. Otherwise, \(c_1x_1 + c_2x_2 + c_3 = 0\) must be a separation plane for \(P\). See the next exercise for a proof.

As a remark, for the above instance, LP will return \(c_4 = \infty\).
Problem 4. Prove that the reduction from linear classification to linear programming explained on Slide 21 of the notes of Lecture 4 is correct.

Proof. We will use the notations on Slide 21. We will prove that $P$ is linear separable if and only if the $c_{d+1}$ returned by LP is strictly greater than 0.

The If-Direction. We claim that in this case $c_1 x_1 + c_2 x_2 + ... + c_d x_d = 0$ is a separation plane for $P$. Indeed, if $p(x_1, ..., x_d)$ is red, we know (from the result of LP) that $c_1 x_1 + c_2 x_2 + ... + c_d x_d \geq c_{d+1} > 0$. Similarly, if $p(x_1, ..., x_d)$ is blue, we know that $c_1 x_1 + c_2 x_2 + ... + c_d x_d \leq -c_{d+1} < 0$.

The Only-If Direction. Since $P$ is linearly separable, there is a plane $\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d = 0$ such that
- if $p(x_1, ..., x_d)$ is red, $\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d > 0$
- if $p(x_1, ..., x_d)$ is blue, $\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d < 0$.

Define $\gamma = \min_{p \in P} |\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d|$. We thus know that $\gamma$ is strictly greater than 0. It thus follows that
- if $p(x_1, ..., x_d)$ is red, $\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d \geq \gamma$
- if $p(x_1, ..., x_d)$ is blue, $\alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_d x_d \leq -\gamma$.

Therefore, $c_1 = \alpha_1, c_2 = \alpha_2, ..., c_d = \alpha_d, c_{d+1} = \gamma$ satisfy all the inequalities on Slide 21. As the LP needs to maximize $c_{d+1}$, we thus know that the $c_{d+1}$ eventually returned by LP must be at least $\gamma > 0$. \qed

Problem 5. The figure below shows the boundary lines of 5 half-planes. Consider the execution of the linear programming algorithm we discussed in the class on these 5 half-planes. Recall that the algorithm starts by randomly permuting the boundary lines, and assume that the resulting permutation is exactly $\ell_1, \ell_2, ..., \ell_5$. The algorithm then processes $\ell_i$ in the $i$-th round, for $i = 1, ..., 5$, and at any moment maintains a point $p$ as the current answer. Explain which point $p$ is at the end of each round, starting from $i = 2$.  

![Diagram of boundary lines and points](image)
Answer.
Let $H_1, \ldots, H_5$ be the half-planes whose boundary lines are $\ell_1, \ldots, \ell_5$, respectively. At the end of the second round, $p$ is the intersection $A$ of $\ell_1$ and $\ell_2$. At Round 3, the algorithm checks whether $p = A$ satisfies $H_3$. Since the answer is yes, the algorithm does not change $p$, and moves on to the next round. In Round 4, $p = A$ is tested against $H_4$. As $p$ does not fall in $H_4$, the algorithm computes a new $p$ as the leftmost point on $\ell_4$ that satisfies all of $H_1, \ldots, H_4$. As a result, $p$ is set to $B$. Finally, the last round processes $H_5$. As $p = B$ falls in $H_5$, the algorithm finishes with $B$ as the final answer.