# Homework 02

**Due:** January 28, 9PM
**Point total:** 60

**Instructions:**

- Submit your PDF and .py file to Blackboard by the due date and time. Please do not zip your files together, as this interferes with Blackboard's preview functionality. Always show all your work, and for full credit, you must use the method that the problem instructs you to use (unless none is mentioned). Handwritten or typeset solutions are both acceptable, but unreadable submissions will be penalized. You may discuss problems with other students, but you may not write up solutions together, copy solutions from a common whiteboard, or otherwise share your written work or code. Do not use code or language that is copied from the Internet or other students; attribute the ideas *and* rephrase in your own words.

## Problem 1 (12 points, 4 each)

In each case, find the matrix that performs the transformation described. (Note that you are looking for a *single* matrix, which you can find by multiplying out matrices that perform each individual operation. Be careful about which matrix goes first or last.) You can assume the direction of rotation is the one that is most convenient for our formulas.

**i.** In two dimensions: Swap the $x$ and $y$ coordinates, then rotate 30 degrees about the origin.

*Solution:* $\begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix}$

**ii.** In three dimensions: double the length, then rotate 45 degrees around the x axis, then 45 degrees around the z axis

*Solution:* $\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{3}/2 & -1/2 \\ 0 & 1/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -1/2 & 1/2 \\ \sqrt{2}/2 & 1/2 & -1/2 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$

**iii.** In four dimensions: Project down to three dimensions ($xyz$) by removing the $w$ coordinate (and producing a three-dimensional vector), then rotate 60 degrees around the $y$-axis. (For the projection matrix, notice that the matrix needs to "shift" the elements to occupy new places in the smaller vector, since $w$ came first but is now dropped.)

*Solution:* $\begin{bmatrix} 1/2 & 0 & \sqrt{3}/2 \\ 0 & 1 & 0 \\ -\sqrt{3}/2 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1/2 & 0 & \sqrt{3}/2 \\ 0 & 0 & 1 & 0 \\ 0 & -\sqrt{3}/2 & 0 & 1/2 \end{bmatrix}$

## Problem 2 (10 points)

In each subproblem, you are given three vectors, $\vec{q}, \vec{r}$, and $\vec{s}$. Determine whether the three vectors are linearly independent. If they are not, find the dimension of their span.

**i.** $\vec{q} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \vec{r} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \vec{s} = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$

*Solution:* Not linearly independent; dimension of the span is 2 (verifiable with Gauss-Jordan).

**ii.** $\vec{q} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \vec{r} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \vec{s} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$

*Solution:* Not linearly independent; dimension of the span is 2 (again verifiable with Gauss-Jordan). While these might seem more redundant, as long as one vector isn't a multiple of another, there are still at least two dimensions in the span.

**iii.** $\vec{q} = \begin{bmatrix} \pi \\ 1 \\ 1 \end{bmatrix}, \vec{r} = \begin{bmatrix} -1 \\ \pi \\ 0 \end{bmatrix}, \vec{s} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$

*Solution:* Linearly independent (provable with Gauss-Jordan elimination).

## Problem 3 (8 pts, 4 each)

**i.** Find an example of four 4-dimensional vectors such that any three of them are linearly independent, but all four together are not. (Hint: You can do this with vectors of just zeros and ones.) Explain how you know how all four are linearly dependent, and explain how you know that any three of them are linearly independent.

*Solution:* An example would be the vectors $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$. All four together are clearly dependent, because the fourth is the sum of the first three. The first three alone are independent because they're standard basis vectors (and clearly none is the sum of the others). Any set of three that includes the last vector must also be linearly independent, because the two standard basis vectors are definitely independent from each other, and the last vector always provides a component that neither of the other two can provide.

**ii.** A student has proposed a strange variant of a perceptron in which the weights are always altered by $\alpha\vec{1}$ or $-\alpha\vec{1}$, where $\vec{1}$ is the all ones vector and $\alpha$ is a small constant. (This is instead of using the input $\vec{x}$ to calculate the adjustment to the weights.) What is the dimension of the span of these vectors? How does the dimension of the span lead us to conclude that this perceptron can't possibly learn all possible decision boundaries for $n$-dimensional input?

*Solution:* The span has dimension 1, since one is just a constant multiplied by the other. Since the full space of possible perceptron boundaries has $n + 1$ dimensions (counting the bias), this can't possibly represent all possible decision boundaries.

## Problem 4 (9 points, 3 each)

Try going through the perceptron tutorial (`perceptron_fish_tutorial.ipynb`) before answering these questions.

**i.** In the provided demo code (with 20 epochs), how many total times is a perceptron classification compared to a true classification?

*Solution:* Since there are 20 epochs, 100 fish per group, and 2 groups, that's 4000 classifications compared to true classifications.

**ii.** Why do we see no decision boundary after the first epoch (and before the second)? Justify your explanation with an equation for the decision boundary in $y = mx + b$ form (round to two digits after the decimal place).

*Solution:* We can't see the line because it's outside the viewbox. Specifically, the decision boundary is $1 + 7.17x + 2.13y = 0$, and rearranging those terms gives $y = -3.37x - 0.47$. That's a negative y-intercept and a negative slope.

**iii.** If a perceptron has no bias term (weight for a constant input), is it possible that an incorrect classification of a point could result in no update to the weights? If so, describe what input(s) lead to that situation. If not, explain why not.

*Solution:* Yes, with no bias term, the zero vector will have this effect. Any other vector will have a nonzero dot product with itself, and cause an update.

### Problem 5 (21 points)

This problem is contained in `hw2_perceptron_transform.ipynb`. Please submit your work as a .py file along with a PDF for the other problems. (Don't submit a whole Python notebook, please.)