1. (Programming) RANSAC is widely used in fitting models from sample points with outliers. Please implement a program to fit a straight 2D line using RANSAC from the following sample points:
(-2, 0), (0, 0.9), (2, 2.0), (3, 6.5), (4, 2.9), (5, 8.8), (6, 3.95), (8, 5.03), (10, 5.97), (12, 7.1), (13, 1.2), (14, 8.2), (16, 8.5) (18, 10.1). Please show your result graphically.

2. (Programming) AdaBoost is a powerful classification tool, with which a strong classifier can be learned by composing a set of weak classifiers. In our lecture, we use a vivid example to demonstrate the basic idea of AdaBoost. Now, your task is to implement this demo.

Training:
There are 10 samples on a 2-D plane and information of the $i$th sample is given as $(x_i, y_i, l_i)$, where $(x_i, y_i)$ is its coordinate and $l_i$ is its label. 10 samples are (80, 144, +1), (93, 232, +1), (136, 275, -1), (147, 131, -1), (159, 69, +1), (214, 31, +1), (214, 152, -1), (257, 83, +1), (307, 62, -1), (307, 231, -1). Weak classifiers are vertical or horizontal lines as described in our lecture. The final trained strong classifier actually is a function having the form,

\[ \text{Label} = \text{strongClassifier}(x, y) \]
Finally, test your resultant strong classifier to verify whether it can correctly classify all the training samples.

3. (Math) There are \( n \) \( p \)-dimensional data points and we can stack them into a data matrix, \( \mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n, \mathbf{x}_i \in \mathbb{R}^p, \mathbf{X} \in \mathbb{R}^{p \times n} \)

The covariance matrix of \( \mathbf{X} \) is \( \mathbf{C} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \), where \( \mu = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \) (actually, it is the mean of the data points)

Based on discussions in our lecture, we know that if \( \alpha_1 \) is the eigen-vector associated with the largest eigen-value of \( \mathbf{C} \), the data projections along \( \alpha_1 \) will have the largest variance.

Now let’s consider such an orientation \( \alpha_2 \). It is orthogonal to \( \alpha_1 \); and among all the orientations orthogonal to \( \alpha_1 \), the variance of data projections to \( \alpha_2 \) is the largest one.

Please prove that: \( \alpha_2 \) actually is the eigen-vector of \( \mathbf{C} \) associated to \( \mathbf{C} \)’s second largest eigen-value. (we can assume that \( \alpha_2 \) is a unit-vector)

4. (Math) In our lecture, we mentioned that for logistic regression, the cost function is,

\[
J(\theta) = -\sum_{i=1}^m y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))
\]

Please verify that the gradient of this cost function is

\[
\nabla_\theta J(\theta) = \sum_{i=1}^m x_i (h_\theta(x_i) - y_i)
\]
5. **(Programming)** In intelligent retail, one task is to investigate the proportion of each commodity occupying shelves. In this assignment, suppose that you are provided a surveillance video of a shelf and you need to recognize and locate two specific kinds of products, “康师傅香辣牛肉面” and “康师父卤香牛肉面” in real time. You can use YoloV2, YoloV3, Mobile-SSD, or pelee-SSD (object detection approaches) for this task.

The test video is given on the course website. Please show your results to TA.