HOMEWORK 4: BOOSTING, K-MEANS, GMMS AND EM

10-701 Introduction to Machine Learning
(PhD) (Fall 2020)
Carnegie Mellon University
piazza.com/cmu/fall2020/10701/home
OUT: October 21st, 2020*
DUE: November 6th, 2020 11:59 PM
TAs: John Grace, Bhuvan Agrawal

START HERE: Instructions

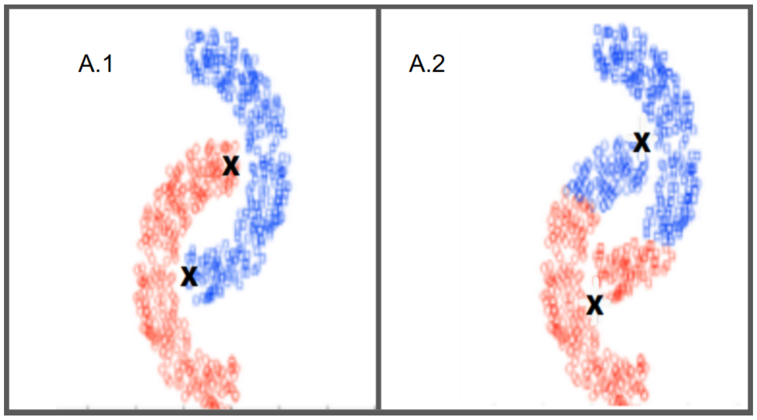
- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only. See the Academic Integrity Section on the course site for more information: https://www.cs.cmu.edu/~epxing/Class/10701-20/about.html#academic-integrity-policies
- **Late Submission Policy:** See the late submission policy here: https://www.cs.cmu.edu/~epxing/Class/10701-20/about.html#late-homework-policy
- **Submitting your work:**
  - **Gradescope:** There will be two submission slots for this homework on Gradescope: Written and Programming.
    For the written problems such as short answer, multiple choice, derivations, proofs, or plots, we will be using the written submission slot. Please use the provided template. The best way to format your homework is by using the Latex template released in the handout and writing your solutions in Latex. However submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Each derivation/proof should be completed in the boxes provided below the question, **you should not change the sizes of these boxes** as Gradescope is expecting your solved homework PDF to match the template on Gradescope. If you find you need more space than the box provides you should consider cutting your solution down to its relevant parts, if you see no way to do this it please add an additional page a the end of the homework and guide us there with a 'See page xx for the rest of the solution'.
    You are also required to upload your code, which you wrote to solve the final question of this homework, to the Programming submission slot. Your code may be ran by TAs so please make sure it is in a workable state.
    Regrade requests can be made after the homework grades are released, however this gives the

*Compiled on Saturday 24th October, 2020 at 19:20

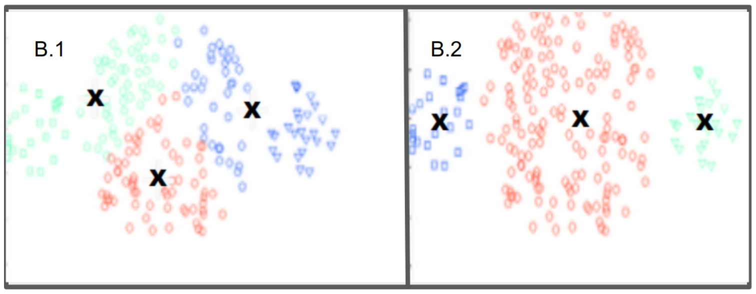
TA the opportunity to regrade your entire paper, meaning if additional mistakes are found then points will be deducted.

For multiple choice or select all that apply questions, shade in the box or circle in the template document corresponding to the correct answer(s) for each of the questions. For LaTeXusers, use ■ and ●for shaded boxes and circles, and don't change anything else.

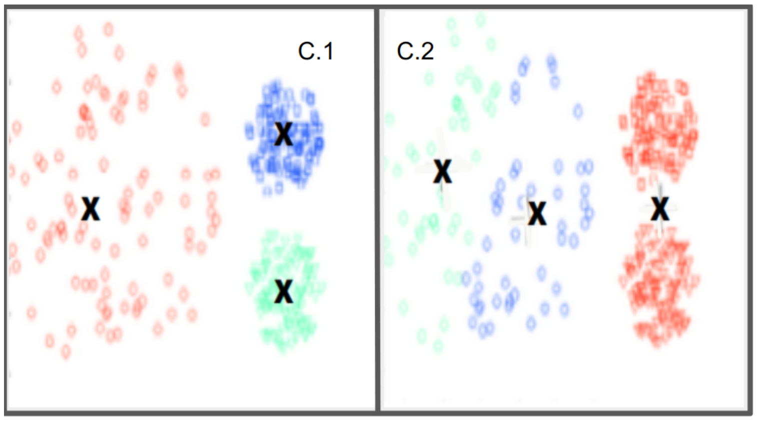
1 K Means [24pts]



(a) Dataset A



(b) Dataset B



(c) Dataset C

Figure 1.1: Datasets

1. **[3pts]** Consider the 3 datasets A, B and C as shown in Figure 1.1. Each dataset is classified into k clusters as represented by different colors in the figure. For each dataset, determine which image with cluster centers (denoted by X) is generated by K-means method.The distance measure used here is the Euclidean distance.

   1.1. **[1pt]** Dataset A **(Select one)**

   ◯ A.1

   ◯ A.2

   1.2. **[1pt]** Dataset B **(Select one)**

   ◯ B.1

   ◯ B.2

   1.3. **[1pt]** Dataset C **(Select one)**

   ◯ C.1

   ◯ C.2

2. **[10pts]** Consider a Dataset **D** with 5 points as shown below. Perform a k-means clustering on this dataset with k as 2 using the Euclidean distance as the distance function. Remember that in the K-means algorithm, an iteration consists of performing following tasks: Assigning each data point to it's nearest cluster center followed by recomputation of those centers based on all the data points assigned to it. Initially, the 2 cluster centers are chosen randomly as $\mu0$ = (5.3, 3.5) (0), $\mu1$ = (5.1, 4.2) (1).

$$D = \begin{bmatrix} 5.5 & 3.1 \\ 5.1 & 4.8 \\ 6.3 & 3.0 \\ 5.5 & 4.4 \\ 6.8 & 3.5 \end{bmatrix}$$

   2.1. **[3pts]** What will be the coordinates of the center for cluster 0 after the first iteration?

   2.2. **[3pts]** What will be the coordinates of the center for cluster 1 after the first iteration?

   2.3. **[2pt]** How many points will belong to cluster 0 after the first iteration?

   2.4. **[2pt]** How many points will belong to cluster 1 after the first iteration?

3. **[6pts]** Recall that in k-means clustering we attempt to find k cluster centers $c_j \in \mathbb{R}^d, j \in \{1, \dots, k\}$ such that the total distance between each datapoint and the nearest cluster center is minimized. Then the objective function is,

$$\sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} ||x_i - c_j||^2 \tag{1.1}$$

In other words, we attempt to find $c_1, \dots, c_k$ that minimizes Eq. (1.1), where n is the number of data points. To do so, we iterate between assigning x_i to the nearest cluster center and updating each cluster center c_j to the average of all points assigned to the j th cluster. Instead of holding the number of clusters k fixed, your friend John tries to minimize Eq. (1.1) over k. Yet, you found this idea to be a bad one.

Specifically, you convinced John by providing two values α, the minimum possible value of Eq. (1.1), and β, the value of k when Eq. (1.1) is minimized.

3.1. **[3pts]** What is the value of $\alpha + \beta$ when $n = 100$?

3.2. **[3pts]** We want to see how k-means clustering works on a single dimension. Consider the case in which k = 3 and we have 4 data points $x_1 = 1, x_2 = 2, x_3 = 5, x_4 = 7$. What is the optimal value of the objective Eq. (1.1)?

1.1 K-Means [5 pts]

Given a set of n observations X, and cluster centers $\mu_1, ..., \mu_k$, assign each observation to its closest center. So now we can write each data point as x_{ij}, which is the jth point in the ith cluster.

We try to minimize

$$SD_K = \sum_{i=1}^{k} SD_{K_i}$$

where

$$SD_{K_i} = \sum_{j=1}^{m_k} ||x_{ij} - \mu_i||^2$$

Prove that

$$\min_{\mu_1,...,\mu_k} SD_K$$

is a non-increasing function of k.

2 Gaussian Mixture Models [20 pts]

In this section, you will study the update rules for Gaussian Mixture Models (GMMs) using the expectation-maximization (EM) algorithm.

Remember that the general EM algorithm has two steps:

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}}\left[\log L(\theta; X, Z)\right]$$

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

X here is the dataset $\{x^{(i)}\}_{i=1}^{n}$ where $x^{(i)} \in \mathbb{R}^d$. We use slack variables $\{z^{(i)}\}_{i=1}^{n}$ to complete our proof. Each $z^{(i)}$ can take values $1, \ldots, k$. For simplicity, we use $z_k^{(i)}$ to denote the event that $z^{(i)} = k$, which means that the ith data point is generated by the kth Gaussian mixture. We use θ to represent the mixture mean, covariance, and prior probability $(\mu, \Sigma, \tilde{\pi})$ for simplicity. Assume the covariance matrices are invertible.

2.1 E-step [8 pts]

1. **[4 pts]** Derive with steps the expression of $\mathbb{P}(x^{(i)}, z_k^{(i)}|\theta^{(t)})$ and $\mathbb{P}(z_k^{(i)}|x^{(i)}, \theta^{(t)})$.

2. **[4 pts]** In the following questions, use $\eta^{(t)}(z_k^{(i)})$ to represent $\mathbb{P}(z_k^{(i)}|x^{(i)}, \theta^{(t)})$ for simplicity.

   Derive with steps the expression of $Q(\theta|\theta^{(t)})$.

2.2 M-step [12 pts]

1. **[4 pts]** Let $N_k^{(t)} = \sum_{i=1}^n \eta^{(t)}(z_k^{(i)})$ in the following questions. Show with steps that

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n \eta^{(t)}(z_k^{(i)}) x^{(i)}}{N_k}.$$

2. **[4 pts]** Show with steps that

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^n \eta^{(t)}(z_k^{(i)})(x^{(i)} - \mu_k^{(t+1)})(x^{(i)} - \mu_k^{(t+1)})^T}{N_k^{(t)}}.$$

You may want to use the fact that $\frac{d|A|}{dA} = |A|A^{-T}$, where $|A| = det(A)$ is the determinant of A.

Additionally, we have $\frac{dx^T Ax}{dA} = xx^T$ and $\frac{dx^T A^{-1} x}{dA} = -A^{-T} xx^T A^{-T}$.

3. **[4 pts]** Show with steps that

$$\tilde{\pi}_k^{(t+1)} = \frac{N_k^{(t)}}{n}.$$

Hint: You may want to use Lagrange multipliers.

3 EM [20 pts]

Suppose the crew of a spaceship is interested in designing an algorithm that takes in certain features of a crewmate, such as height, weight, etc. and predicts whether they are the impostor or not. Suppose all such features are represented in vector X but they do not have access to a subject's suspiciousness (i.e., suspicious or not-suspicious), which they represent by the variable $Z \in \{0, 1\}$. Crewmate Yellow wants to come up with a model to predict whether a crewmate is an imposter ($y = 1$) or not ($y = 0$), using this feature vector X. To do so Crewmate Yellow will be using the EM algorithm, and they will assume a logistic regression model to represent the relation between suspiciousness and all the features X, i.e.

$$\sigma_z(X) = p(Z = 1|X) = \frac{\exp(w_z^T X + b_z)}{1 + \exp(w_z^T X + b_z)}$$

where w_z and b_z are the parameters of the logistic function. Additionally Crewmate Yellow assumes Y given X and Z follows a logistic regression distribution based on the value of Z. So $\sigma_{y0} = p(Y = 1|X, Z = 0)$ is one logistic function with parameters w_{y0} and b_{y0} as its parameters and $\sigma_{y1} = p(Y = 1|X, Z = 1)$ is another logistic function with w_{y1} and b_{y1} as its parameters.

1. **[3 pts]** Suppose a dataset $\mathcal{D} = \{(X_1, y_1), ..., (X_n, y_n)\}$ is given. Derive the joint conditional likelihood of the data (joint between Z and Y) conditioned on X, i.e. $p(Y, Z|X, \Theta)$ and show all your steps clearly. Condense your formulation to a single representation similar to the likelihood for logistic regression (hint: use the binary variables as powers) in terms of parameters introduced above. You can use $\Theta = (w_{y0}, b_{y0}, w_{y1}, b_{y1}, w_z, b_z)$ if that shortens your expressions in any part of your derivations.

2. **[2 pts]** Write the $Q(\Theta|\Theta_t)$ function from the algorithm for this setting. You have to write your answer in terms of all the datapoints $\{x_i, y_i, z_i\}_{i=1}^{n}$, Θ and the corresponding probability functions. You don't have to expand out your answer in terms of σ_z, σ_{y0}, and σ_{y1}.

3. **[5 pts]** Derive the conditional posterior (conditioned on X and y) that would be useful for the Expectation step of EM algorithm for a single datapoint i and show all your steps clearly. This should only be based on σ_{y0} ,σ_{y1} and σ_z introduced above.

4. **[10 pts]** Suppose Crewmate Red tells Cremate Yellow that the suspiciousness of a crewmate and the fact that they are the impostor are independent given the features of the crewmate, i.e. Y actually does not depend on Z given X. In such a case Crewmate Yellow can model $p(Y|X)$ using the logistic regression model directly. Suppose Crewmate Yellow incorporates the independence of Y from Z given X while applying the EM algorithm. Show that the part of the solution of the EM algorithm which explains $p(Y|X, Z, \Theta)$ will indeed give the maximizing parameters of the log-likelihood for $p(Y|X)$ based on logistic regression model.

4 Miscellaneous [16 pts]

1. **[1 pts] True or False:** The final decision boundary when using Adaboost with decision stumps (one-level decision trees) as the weak classifiers is linear.

   ◯ True

   ◯ False

2. **[1 pt] True or False:** The general EM algorithm converges to the global maximum.

   ◯ True

   ◯ False

3. **[4 pts] Select all that apply:** Suppose you have observed data $X = \{x_i\}_{i=1}^n$ along with the unobserved hidden variable $Z = \{z_i\}_{i=1}^n$ where $z_i \in [K]$. The model parameters are denoted by θ with $\theta^{(t)}$ denoting the estimate at step t. Using the notation from class, we wish to maximize the incomplete log-likelihood $\ell(\theta; X) = \log p(X|\theta)$. Which of the following are valid lower-bounds (free energy) to the incomplete log-likelihood $F(q, \theta) \leq \ell(\theta; X)$?

   ☐ $\sum_{i=1}^n \log p(x_i|\theta)$

   ☐ $\sum_{i=1}^n \log \left(\sum_{k=1}^K p(x_i, z_i = k|\theta) \right)$

   ☐ $\sum_{i=1}^n \sum_{k=1}^K p(z_i = k|x_i, \theta^{(t)}) \log \frac{p(x_i, z_i = k|\theta)}{p(z_i = k|x_i, \theta^{(t)})}$

   ☐ $\frac{1}{K} \sum_{i=1}^n \sum_{k=1}^K \log p(x_i, z_i = k|\theta) + n \log K$

   ☐ None of the above

4. **[2 pts] Select one:** Recall the K-means objective:

$$F(\boldsymbol{\mu}, C) = \sum_{j=1}^{n} \|\boldsymbol{\mu}_{C(j)} - \boldsymbol{x}_j\|_2^2 = \sum_{i=1}^{K} \sum_{j \in N_i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_j\|_2^2$$

where $\{\boldsymbol{x}_j\}_{j=1}^n$ are the datapoints, $\{\boldsymbol{\mu}_i\}_{i=1}^k$ are the centers and C denotes the point allocations. $N_i = \{j \in [n] \mid C(j) = i\}$ is the set of indices of points that have been assigned to cluster i. Suppose we change the objective to use the L_1 distance instead of the L_2 distance, like so:

$$\hat{F}(\boldsymbol{\mu}, C) = \sum_{j=1}^{n} \|\boldsymbol{\mu}_{C(j)} - \boldsymbol{x}_j\|_1 = \sum_{i=1}^{K} \sum_{j \in N_i} \|\boldsymbol{\mu}_i - \boldsymbol{x}_j\|_1$$

Keeping the point allocations $C^{(t)}$ fixed, what are the new estimates of the centers $\boldsymbol{\mu}_i^{(t+1)}$ under this new objective? Here, μ_{ik} and x_{jk} denote the k-th component of $\boldsymbol{\mu}_i$ and $\boldsymbol{x}_j$ respectively.

○ $\mu_{ik}^{(t+1)} = \frac{1}{|N_i^{(t)}|} \sum_{j \in N_i^{(t)}} x_{jk}$

○ $\mu_{ik}^{(t+1)} = \frac{1}{|N_i^{(t)}|} \sum_{j \in N_i^{(t)}} |x_{jk}|$

○ $\mu_{ik}^{(t+1)} = \text{Median}_{j \in N_i^{(t)}} \{x_{jk}\}$

○ $\mu_{ik}^{(t+1)} = \sqrt{\frac{1}{|N_i^{(t)}|} \sum_{j \in N_i^{(t)}} x_{jk}^2}$

5. **[Bernoulli Mixture Models]** There are two coins $\mathcal{C}_1$ and $\mathcal{C}_2$ with unknown probabilities of heads, denoted by p and q respectively. $\mathcal{C}_1$ is chosen with probability π and $\mathcal{C}_2$ is chosen with probability $1-\pi$. The chosen coin is flipped once and the result is recorded. The experiment is repeated five times and the final result is recorded as $X = \{H, H, T, H, T\}$. You wish to use the EM algorithm to estimate the parameters $\theta = [p, q, \pi]$. Suppose Z denotes the hidden variable where $z_i = 1$ if $\mathcal{C}_1$ was tossed for the i-th flip and 0 if $\mathcal{C}_2$ was tossed. You start off with an initial guess of $p^{(0)} = 1/4, q^{(0)} = 2/3$, and $\pi^{(0)} = 1/2$.

In the following questions, please write your answer as an **exact irreducible fraction** $\frac{a}{b}$ where a, b are co-prime integers. For example, if your answer is 0.4125, please write it as $\frac{33}{80}$. The horizontal bar for the fraction $\frac{\square}{\square}$ is already present in the answer box.

(a) **[1 pt] E step:** Compute $p(z_i = 1 | x_i = H, \theta^{(0)})$

$$\boxed{\frac{\quad}{\quad}}$$

(b) **[1 pt] E step:** Compute $p(z_i = 1 | x_i = T, \theta^{(0)})$

——

(c) **[2 pt] M step:** Compute $\pi^{(1)}$ Hint: $\theta^{(t+1)} = \text{argmax}_\theta \, \mathbb{E}_{Z|X,\theta^{(t)}}[\log p(X, Z|\theta)]$

——

(d) **[2 pt] M step:** Compute $p^{(1)}$

——

(e) **[2 pt] M step:** Compute $q^{(1)}$

——

5 Programming: AdaBoost Algorithm [17 pts]

In this section, you will implement the AdaBoost algorithm with decision stumps, i.e., 1-level decision trees. You will need to implement the adaboost algorithm from scratch and train and test the classifier on the dataset provided.

The 1-level decision stumps are just binary questions asking whether the point is greater or less than a particular dividing line–the dividing line can be horizontal or vertical.

Also, each potential dividing line for the decision stumps must come from the training data itself. So when you are looking for a good separator, you should iterator through your data points in order to look for one.

We recommend designing your code in the following way. However you can write your program according to your design flow.

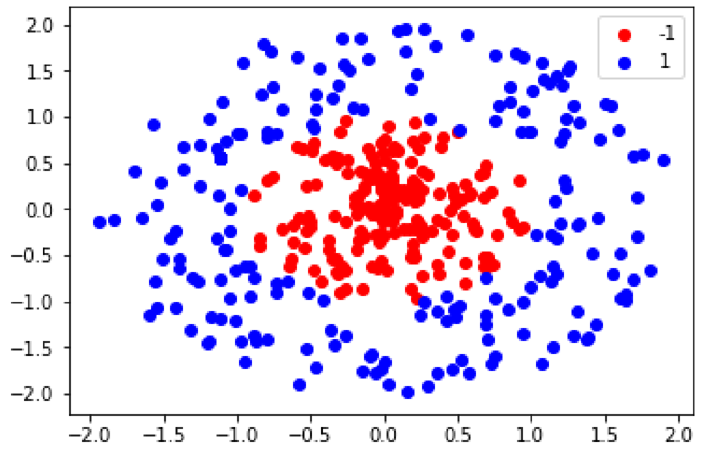


Figure 5.1: Example dataset. Each point $\mathbf{x} \in [-2, 2] \times [-2, 2]$ and $y \in \{-1, 1\}$

The dataset for this task is synthetically generated and has two entries $\mathbf{x}_i = (x_1,\ x_2)$ and y. Here $\mathbf{x}_i = (x_1, x_2) \in [-2, 2] \times [-2, 2]$ is the *i-th* instance and $y_i \in \{-1, 1\}$ is its corresponding label. The training and test dataset are in "train_adaboost.csv" and "test_adaboost.csv" respectively.

1. **Suggested structure for the programming assignment**: Consider writing the following functions:
   (a) **read_data**: This function reads the data from the input file.
   (b) **weak_classifier**: This function finds the best weak classifier, which is a 1-level decision tree. Defining n as number of training points and d as the number of features per data point, the inputs to the function should be the input data ($n \times d$ array), the true labels ($n \times 1$ array) and the current distribution D ($n \times 1$ array) and you should return the best weak classifier with the best split based on the error. Some of the things to include in the output can be: best feature index,best split value, label, value of β_t and predicted labels by the best weak classifier. Note: β_t should be a ($T \times 1$ array, for $t = 1 \ldots T$, and T denotes the number of iterations that you choose to run the boosting algorithm)
   (c) **update_weights**: This function computes the updated distribution D_{t+1}, The inputs to this function should be current distribution D ($n \times 1$ array), the value of β_t, the true target values ($n \times 1$

array) and the predicted target values ($n \times 1$ array). And the function should output the updated distribution D.

(d) **adaboost_predict**: This function returns the predicted labels for each weak classifier. The inputs: the input data ($n \times d$ array), the array of weak classifiers ($T \times 3$ array) and the array of β_t for $t = 1 \ldots T$ ($T \times 1$ array) and output the predicted labels ($n \times 1$ array)

(e) **eval_model**: This function evaluates the model with test data by measuring the accuracy. Assuming we have m test points, the inputs should be the test data ($m \times d$ array), the true labels for test data ($m \times 1$ array), the array of weak classifiers ($T \times 3$ array), and the array of β_t for $t = 1 \ldots T$ ($T \times 1$ array). The function should output: the predicted labels ($m \times 1$ array) and the accuracy.

(f) **Adaboost_train**: This function trains the model by using AdaBoost algorithm.

Inputs:
- The number of iterations (T)
- The input data for training data ($n \times d$ array)
- The true labels for training data ($n \times 1$ array)
- The input features for test data ($m \times 2$ array)
- The true labels for test data ($m \times 1$ array)

Output:
- The array of weak classifiers ($T \times 3$ array)
- The array of β_t for $t = 1 \ldots T$ ($T \times 1$ array)

(g) **main**:

```
def main():
    num_iter = 400

    X_train, y_train = read_data("train_adaboost.csv")
    X_test, y_test = read_data("test_adaboost.csv")

    hlist, alphalist = train(num_iter, X_train, y_train,
    X_test, y_test)
    final_pred, final_acc = eval_model(X_test, y_test,
    hlist, alphalist)
```

Details on Implementation:

1. Start with a base learner function that takes in training data and weights of each sample, and outputs the best decision stump under the current AdaBoost distribution D_t, e.g. the feature index that your decision stump is separating (x_1 or x_2), the position of the decision stump, and to which side your decision stump will label as **1**.
2. Your preliminary prediction might belongs to $\{0, 1\}$, in order to transform it to **+1, -1** you can consider using the following suggestion:

$$prediction = preliminary\_prediction * 2 - 1, \quad prediction \in \{-1, +1\}$$

1. **[5pts]** Test your classifier by training with varying the number of iteration(num_iter) from 1 to 50. Make a plot of the test accuracy versus number of iterations from 1 to 50. Be sure to label your graph sufficiently.

2. **[3pts]** What is the test accuracy after 50 iterations?

3. **[9pts]** What is the best separator for the first three iterations? Please say whether it is a horizontal or vertical line, and the coordinate of the line. (e.g Horizontal at 0.5)

Iteration 1:

Iteration 2:

Iteration 3:

6 Collaboration Questions

1. (a) Did you receive any help whatsoever from anyone in solving this assignment?

   (b) If you answered 'yes', give full details (e.g. "Jane Doe explained to me what is asked in Question 3.4")

2. (a) Did you give any help whatsoever to anyone in solving this assignment?

   (b) If you answered 'yes', give full details (e.g. "I pointed Joe Smith to section 2.3 since he didn't know how to proceed with Question 2")

3. (a) Did you find or come across code that implements any part of this assignment?

   (b) If you answered 'yes', give full details (book & page, URL & location within the page, etc.).