1. Supervised Learning [6]

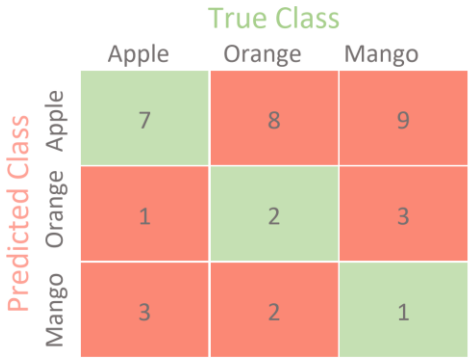
a) Assume you learn a decision tree for a dataset D and you observe overfitting. What things could done to reduce overfitting? [3]

a. create more examples for D [1.5]

b. reduce the number of nodes of the decision tree [1.5]

b) A confusion Matrix of a classification model for distinguishing apples from oranges and mangos is given below:

What is the accuracy of the classification model; what is its precision for class Apple; what is its recall for the class Apple? It is okay to represent your answers as fractions; e.g. 17/36! [3]



Accuracy: 7+2+1/7+8+9+1+2+3+3+2+1 [1]

Precision Apple: 7/7+8+9 [1]

Recall Apple: 7/7+3+1 [1]

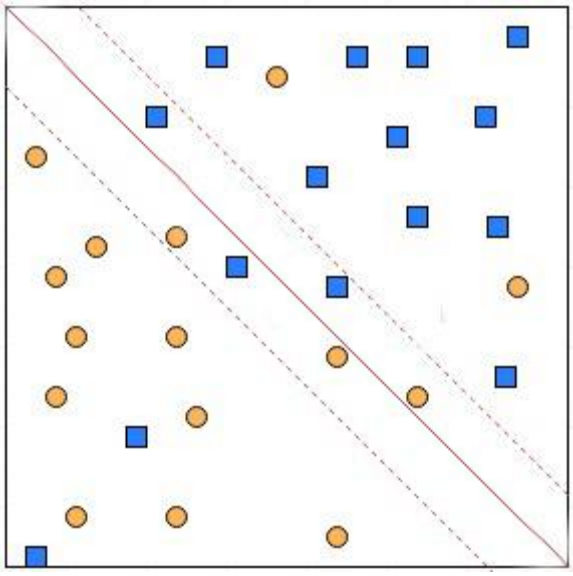
No partial credit!

2) Support Vector Machines [6]

The soft margin support vector machine solves the following optimization problem:

$$\operatorname{argmin} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \qquad \text{subject to } c_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i \quad 1 \leq i \leq n.$$

What does the second term minimize (be precise!)? [1]What is the purpose of C? [1]Next, add arrows to all examples in the figure below, whose ξ_i values are positive---the length of the arrow should correspond with the value to the respective ξ_i ! [2]Are examples with a positive ξ_i value always misclassified; give a reason for your answer! [2]



No answer given at the moment.

3) Clustering [12]

a) Let us assume we run Fuzzy C-means (FCM) for K=2 and the centroid for cluster 1 is (1,1) and the centroid of cluster 2 is (2,3) and hyper parameter p is 2 and we use Manhattan distance; furthermore, point i is: (2,2). Compute the w_{i1} and w_{i2} for point i! [4]

W_{i1}= 1/1**2/(1+1/4)=0.8

W_{i2}=1/2**2)/(1+1/4)=0.2

$$w_{ij} = (1/\boldsymbol{dist}(\boldsymbol{x_i}, \boldsymbol{c_j})^2)^{\frac{1}{p-1}} / \sum_{q=1}^{k} (1/\boldsymbol{dist}(\boldsymbol{x_i}, \boldsymbol{c_q})^2)^{\frac{1}{p-1}}$$

b) Assume we use FCM for 4 points and k=2 and the points and their weights are as follows:

Point 1: (0,0) with w_{11}=1 and w_{12}=0

Point 2: (3,3) with w_{21}=0.7 and w_{22}=0.3

Point 3: (8,9) with w_{31}=0.1 and w_{32}=0.9

Point 4: (12,13) with w_{41}=0 and w_{42}=1

Using the methods FCM uses, compute the centroid of cluster 2; give the formula and its vector. [4]

$Centroid_2$= (1*(12,13)+0.9*(8,9)+0.3*(0.3,0.3)) / 2.2

((12+7.2+0.9/2.2, (13+8.1+0.9)/2.2)=

(20.1/2.2, 22 /2.2)=(9.14,10)

Can give them 2.5 points if they use the correct formula and have a major calculation error and 3 points if they had a minor calculation error; 3.5 points if their answer is (20.1/2.2,22/2/2). At most 1 point, if they use the wrong formula

c) Assume we apply the CLIQUE algorithm to a numerical dataset with attributes A, B, C, D and E. How is CLIQUE different from more traditional clustering algorithms, such as K-means? How does CLIQUE form clusters? [4]

Finds clusters in the subspace rather in the complete space A-B-C-D-E-F [2]
Clusters in subspaces are formed by a growing algorithm which starts with a seed grid-cells which are dense and adds neighboring grid-cells which are dense [2]

Other answers might deserve full or partial credit!

Assessing Performance [7 points]

1. [2 pts] For the bias-variance tradeoff, which of the following substantially increases the test error more than the training error. **(Select one)**

   ○ Bias ● **Variance**

2. [2 pts] A model is considered overfit if it achieves lower training error than another model. **(Select one)**

   ○ True ● **False**

3. [3 pts] The two plots below show 3 points that are drawn randomly from the **same** underlying data generating mechanism:

   $$y_i = f(x_i) + \epsilon,$$

   Suppose we fit a **degree-2 polynomial** to two different datasets drawn from the same process.

   (a) [2 pts] For each plot, draw the parabola that best fits the data.

   (b) [1 pt] Does our chosen model have **low** or **high** variance? Explain in a single sentence.

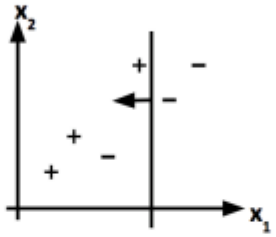
> **Solution:** This model, when using datasets of this size, has **high variance** because we get very different fits for slightly different datasets.

Boosting [7 points]

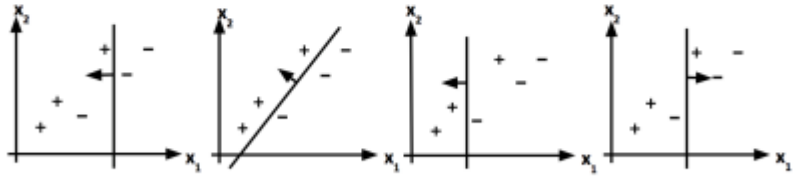
1. [5 points] Consider the algorithm for training the AdaBoost model. Recall that the algorithm trains a series of decision stumps on the data and changes the weights of points it misclassifies.

   The figure below displays a 2-dimensional training dataset, as well as the first stump chosen. The little arrow in the figure is the normal to the stump decision boundary indicating the positive side where the stump predicts +1. All points start with uniform weights.

   (a) [2 pts] **Circle all points** in the figure below whose weight will **increase** as a result of incorporating the first stump.



   (b) [2 pts] Which of the figures below shows the best stump that we could select at the next boosting iteration? **(Select one)**

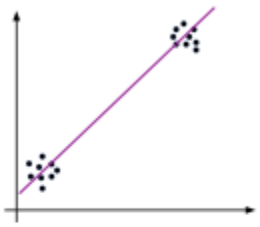


   ○ Fig a ○ Fig b ● **Fig c** ○ Fig d

   (c) [1 pt] If you keep running AdaBoost on this dataset until the training error does not decrease further, the training error will be: **(Select one)**

   ● **0**
   ○ $^1/_6$
   ○ Cannot be determined

1. [4 pts] Suppose you were to use PCA on the following dataset



Dataset for PCA

(a) [2 pts] **On the figure above**, draw the line that is in the direction of the first principal component of the data.

(b) [1 pt] Justify your answer to part (a) in one sentence.

Solution: This is the direction that minimizes reconstruction error (alternatively, the direction of maximum variance).

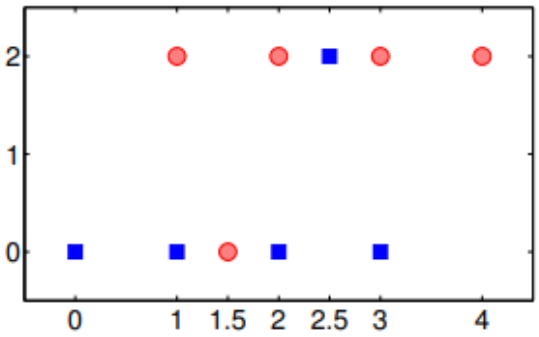
(c) [1 pt] One the line below, draw what the dataset will look like after projecting onto the first principal component.

We are not looking for you to make a picture-perfect projection, but your drawing should capture the general idea of how the data will be projected onto the first principal component.

Problem 5: (8 points) Cross-validation

Suppose that we learn a classifier on the following binary classification data. There are two real-valued features, x_1 and x_2, and a binary class $y \in \{0, 1\}$.

x_1	x_2	y
0	0	0
1	0	0
1.5	0	1
2	0	0
3	0	0
1	2	1
2	2	1
2.5	2	0
3	2	1
4	2	1



We decide to learn a decision tree as described in class. As in class, when the decision tree splits on the real-valued features, it puts the split threshold halfway between the data points on either side of the highest-scoring split. For example, if we first split on x_2, the algorithm would choose to split at $x_2 = 1$, which is halfway between the data at $x_2 = 0$ and $x_2 = 2$.

(a) What is the training error rate of a decision *stump* (decision tree with max depth 1) trained on these data?

(b) What is the training error rate of a full decision tree (no maximum depth) trained on these data?

(c) What is the leave-one-out cross-validation error rate of a decision *stump* (decision tree with max depth 1) trained on these data?

(d) What is the leave-one-out cross-validation error rate of a full decision tree (no maximum depth) trained on these data?