Graded Exercise -1

CS 233 - Introduction to Machine Learning

April 2, 2019

NAME: **SCIPER:**

1 KNN

1. You just moved to apartment 4c of the following 10-story apartment building.

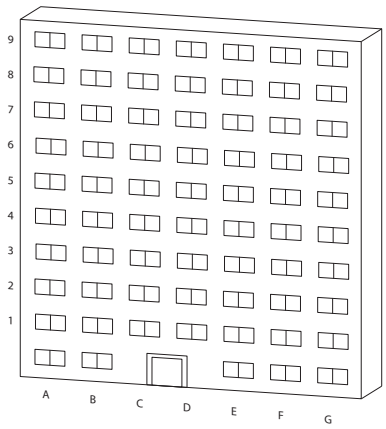

Figure 1: Your new home

You live right in front of the staircase. You don't have internet yet and want to borrow WiFi from your neighbors. Your laptop scans networks and returns the following:

- `Apt8A-gimmeyerpizza`
- `bring_cookies_to_1G_for_pwd`
- `TheCheeseEnthusiastsOf9D`

If you want to *walk* the least to ask your neighbor to lend you their WiFi, what item(s) could you bring to succeed in your mission?

☒ Pizza

☐ Cookies

☒ Cheese

☐ This is Switzerland, we don't talk to our neighbors.

2. Now the above networks are all open. Ignoring the effects of the walls, which network should have the strongest signal?

   ☒ `Apt8A-gimmeyerpizza`

   ☐ `bring_cookies_to_1G_for_pwd`

   ☐ `TheCheeseEnthusiastsOf9D`

   ☐ Equal for cookie and cheese lovers

   ☐ Equal for pizza and cookie lovers

3. In algorithms like perceptron and SVM we often find the expression $\mathbf{w}^T\mathbf{x}$. What is the equivalent of it in the k-NN algorithm:

   ☐ $\mathbf{k}^T\mathbf{x}$

   ☐ The average class of the k nearest neighbors

   ☒ None, because k-NN doesn't find a parametric decision boundary

   ☐ None, because k-NN doesn't have hyperparameters

4. What are the main limitations of the k-NN algorithm:

   ☐ It only works for two and three dimensions

   ☐ It only works on data that is linearly-separable

   ☒ It is inefficient for large datasets (very large N)

   ☐ It only works with Euclidean and Manhattan distance metrics

2 Perceptron

1. The perceptron is defined by a function $y(\mathbf{x}) = f(\mathbf{w}^\top\mathbf{x})$ where $\mathbf{w}$ is a vector of (learned) weights and f is a step function $f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$. Assume we have a perceptron whose decision boundary is defined by the weight vector $\mathbf{w} = [w_0, w_1]$ and which takes a data sample $\mathbf{x} = [x_1, x_2]$ as input. We would like to model a NOR-like logic function defined in the table below. Choose all sets of weights $\mathbf{w}^* = [w_1^*, w_2^*]$ that solve the problem.

   ☐ $\mathbf{w}^* = [0, -1]$

x_1	x_2	y
0	0	1
0	1	-1
1	0	-1
1	1	-1

- ☒ $\mathbf{w}^* = [-2, -0.5]$
- ☐ $\mathbf{w}^* = [-1, 1]$
- ☐ $\mathbf{w}^* = [0, 1]$
- ☐ $\mathbf{w}^*$ does not exist because the perceptron lacks a bias term.

2. When working with perceptron classifier, we often extend each data sample $\mathbf{x} \in \mathbb{R}^D$ by a scalar 1 to obtain $\tilde{\mathbf{x}} \in \mathbb{R}^{D+1}$. This gives us one more trainable parameter, the bias. The weight vector is thus $\tilde{\mathbf{w}} \in \mathbb{R}^{D+1}$. What are the reasons for using a bias term?
   - ☐ Casting the data samples to $D + 1$ dimensions increases the chance of finding a linear decision boundary which might not have existed in D-dimensional case.
   - ☒ Provided that a linear decision boundary exists in a D-dimensional space, the bias term allows the perceptron to always find it regardless of any arbitrary constant translation of all the data samples.
   - ☐ Adding a bias term allows the perceptron to classify the data in more then two classes.
   - ☒ Assume that the original data samples $\mathbf{x}$ were 2-dimensional, which implies that the weight vector $\mathbf{w}$ denotes a decision boundary represented as a line in a 2D space. After adding the bias term, the decision boundary does not have to pass through the origin $[0, 0]$ anymore.

3 Logistic Regression and SVM

1. Which of the following are true about the logistic regression and perceptron algorithms?
   - ☐ The perceptron only works for linearly separable data, whereas logistic regression can handle non-linearly separable data.
   - ☒ The perceptron assigns labels, logistic regression assigns probabilities.
   - ☒ In the perceptron learning algorithm, weights are updated only according to misclassified samples. In logistic regression, we update the weights according to all the samples.
   - ☐ The perceptron algorithm is easily affected by outliers, whereas the logistic regression algorithm is very robust to outliers.

☒ Both the perceptron and logisitic regression are sensitive to outliers whereas SVM is more robust.

4 Higher Dimensional Features and Kernels

1. Given N training vectors of dimension d, where $N = 3, d = 2$, let $\phi : \mathbb{R}^d \to \mathbb{R}^D$ be a feature transform that maps the vectors to a high dimensional space. If ϕ is a polynomial of order 2, what is D? (Hint: Note that polynomial feature expansion involves a bias term.)

   ☐ 3

   ☐ 4

   ☐ 5

   ☒ 6

   ☐ 7

5 Boosting

1. You are given a binary classification problem and you attempt to solve it using logistic regression. Y our classifier performs poorly because your data is not linearly separable.

   Can you use AdaBoost to get a reliable classifier?

   ☒ Yes

   ☐ No

2. In the "update weights" step of the AdaBoost algorithm, the weights of misclassified samples:

   ☒ increase

   ☐ decreases

   ☐ it depends on the data

6 Trees and Forests

1. John wants to know how to do well for the machine learning exam. He collects old statistics and plans to use decision trees to form his model. He now gets 15 data points and 4 features: "whether to stay up late before the exam" (**S**), "whether to attend

all the classes" (**C**), "whether to do all the exercises "(**E**) and "whether to read the textbook "(**T**). We already know the statistics are as below:

$$
\begin{aligned}
Set(all) &= [8+, 7-] \\
Set(\mathbf{S}+) = [4+, 4-], &Set(\mathbf{S}-) = [4+, 3-] \\
Set(\mathbf{C}+) = [6+, 2-], &Set(\mathbf{C}-) = [2+, 5-] \\
Set(\mathbf{E}+) = [7+, 0-], &Set(\mathbf{E}-) = [1+, 7-] \\
Set(\mathbf{T}+) = [6+, 3-], &Set(\mathbf{T}-) = [2+, 4-]
\end{aligned}
$$

Notation: for each number "+" means doing well in the exam, "-" means doing badly in the exam. For each feature, "+" means students follow this feature, "-" means students do not follow this. For example Set(**C**+) = [6+,2-] means for the students who attend all the classes, 6 of them do well in the exam while 2 of them do badly in the exam.

Suppose we are going to use the feature to gain most information at first split, which feature should we choose?

- ☐ whether to stay up late before the exam (**S**)
- ☐ whether to attend all the classes (**C**)
- ☒ whether to do all the exercises (**E**)
- ☐ whether to read the textbook (**T**)