Theoretical Exercises - Week 2

Like last week, it is very imporant that you try to solve every exercise. It is not important that you answer correctly. Spend no more than 5-10 min on each exercise. If you do not solve the exercise, focus on understanding the question, and try to figure out what it is you do not understand.

The TA's will be very happy to answer questions during the TA session or on the board.

Do not despair if you cannot solve them, but try to understand the question and pinpoint which parts you do not understand.

1. Learning Types

In this exercise you must distinguish between Supervised Learning and Unsupervised Learning. Imagine you work at a company that sells *stuff*. The company stores information about its costumers. For each costumer the company saves the following 5 attributes:

`AGE, SEX, INCOME, RESIDENCE, MONEY USED AT COMPANY`

Question 1: In each of the following examples you should determine if the problem is a Supervised or Unsupervised learning problem.

- The company wants to learn how to predict 'MONEY USED AT COMPANY' given 'AGE', 'SEX', 'INCOME' and 'RESIDENCE'. Supervised or Unsupervised?

  Supervised

- The company wants to learn ways of grouping costumers depending on 'AGE'. Supervised or Unsupervised?

  Unsupervised

- The company wants to learn how to predict 'SEX' given 'MONEY SPENT AT COMPANY' and 'AGE'. Supervised or Unsupervised?

  Supervised

- The company wants to target different groups of costumers depending on 'AGE', 'INCOME' and 'MONEY SPENT AT COMPANY'. Supervised or Unsupervised?

  Unsupervised(?)

Question 2: In supervised learning the data is of the form $D_{\text{supervised}} = \{(x_1, y_1), ..., (x_n, y_n)\}$. In unsupervised learning we have data of the form $D_{\text{unsupervised}} = \{x_1, ..., x_n\}$.

Write the form the data would take in each case from Question 1.

HINT: Possible solutions to two of the cases

$$D = \{20 \text{ years}, 21 \text{ years}, 23 \text{ years}, ...\}$$

$$D = \{([100 \text{ kr}, 22 \text{ years}], \text{male}), ([120 \text{ kr}, 30 \text{ years}], \text{female}), ...\}$$

2. Regression Or Classification

Question 1: In each of the following examples you should distinguish between regression and classification.

- In the previous question the company wanted to predict 'MONEY SPENT AT COMPANY' from ('AGE', 'SEX', 'INCOME', 'RESIDENCE'). Is that regression or classification?

- Recognizing the color of wine as white, rose or red. Is that regression or classification?
- Predicting a students grade in machine learning as a function of previous grades (on the 12 scale). Is that regression or classification?
- Predicting email as spam, normal. Regression or classification?

Question 2: In supervised learning we want to approximate an unkown target function $f : X \to Y$. In regression we could have $Y = \mathbb{R}$ and in classification we could have $Y = \{c_1, ..., c_k\}$.

What is Y in the above four cases?

3. The Perceptron

Question 1: Running the Perceptron Learning Algorithm
Assume we are given a training data set with 3 features, of which the first is hardcoded to 1. The data consists of the four examples $((1, 2, 2), 1), ((1, 2, 3), 1), ((1, 4, 2), -1), ((1, 4, 0), -1)$. What hypothesis $w = (w_0, w_1, w_2)$ does it return when initialized with $w = (0, 0, 0)$ and where we always pick the first misclassified point when updating? NOTE: We assume $sign(0) = 0$ and thus is different from all labels.

4. Choosing leaf return value in Decision Trees

(Problem 1.12 from 'Learning From Data')

Given $y_1 \leq \cdots \leq y_n \in \mathbb{R}$ find $h \in \mathbb{R}$ that on average is closest to $y_1, ..., y_n$ measured by squared distance (least squares). That is,

$$h_{\text{mean}} = \arg\min_h \sum_{i=1}^{n} (h - y_i)^2$$

Question 1: Show that $h_{\text{mean}} = \frac{1}{n} \sum_{i=1}^{n} y_i$ is the minimizer.

HINT: Computing the derivative may be worth the time and strain on your brain.

HINT: a local minimum is a global minimum!

Question 2: Consider absolute deviation instead of squared distance, i.e.

$$h_{\text{med}} = \arg\min_h \sum_{i=1}^{n} |h - y_i|$$

Show that $h_{\text{med}} = \text{median}(y_1, ..., y_n)$, the median of the y values is the minimizer.

HINT: Computing derivative may be usefull but $|a|$ is not differentiable at zero but you may set it to zero (ask google about subgradients if you are interested).

HINT: You can also argue purely algorithmically by thinking about what happens with the cost as we sweep h from $-\infty$ to ∞).

HINT: a local minimum is a global minimum!

Question 3: What happens to the solutions $h_{\text{mean}}, h_{\text{med}}$ if we add noise the last element y_n, i.e. $y_n = y_n + \varepsilon$ for $\varepsilon \to \infty$.

Which method is more stable for outliers (data that looks nothing like the remaining data)?

5. Decision Tree Cost with Entropy

In this exercise we examine the entropy-based approach to constructing decision stumps. Recall that, for any leaf ℓ, the entropy in that leaf is $H(\ell) := -\sum_{i=0}^{k-1} p_i \lg_2(p_i)$. Here p_i denotes the fraction of training examples in that leaf having the label i. For binary classification, we thus have $k = 2$.

The entropy of the entire tree T is $H(T) := \sum_\ell (n_\ell/n) H(\ell)$, where ℓ sums over all leaves and n_ℓ is the number of training examples in leaf ℓ.

We consider classification into the $k = 3$ classes Red (0), White (1), Rose (2). We have $n = 9$ training examples. The data has just one feature. The data and labels are as follows:

$$X = \begin{bmatrix} 9. \\ 33. \\ 20. \\ 27. \\ 3. \\ 6. \\ 18. \\ 14. \\ 16. \end{bmatrix}, \quad Y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \end{bmatrix}$$

We consider the split $x < 19$.

Task:

- Compute the entropy of the left and right leaf using this split.
- Compute the entropy of the full tree using this split.

6: Implementing Regression Stumps

In this exercise your task is to implement Regression Trees that consist of one internal node (the root) and two leafs. Such trees are known as Regression Stumps. For the loss/cost function we consider least squares loss $(h(x) - y)^2$

This means that the learning algorithm has to find the best possible feature to split the training data using a single feature value pair in regards to Least Squares loss.

We have decided for you to present a Regression Stump by

- idx: the data/feature vector index to consider in the root node (the one question asked)
- val: the value to compare to for data feature idx in the root node
- left: the value to return for a data point if it ends up in left leaf (x[idx] < val) (only question type we consider in a node)
- right: the value to return for a data point if it ends up in the right leaf (x[idx] >= val)

The approach we follow is as follows. Assume the input data has n data points each a vector of d real numbers.

Basic Algorithm:

For each data feature f:

- Compute for all possible values v for feature f in the training data, the least squares cost of the stump achieved by using feature f and value v in the root using the optimal value in the two leafs. This gives a list of of costs, one for each split (f, v : cost).
- Pick the split f, v with minimal cost and create the corresponding tree by setting idx, val, left, right

Your task is to give a full implementation of this algorithm and specify the running time.

hint: It is fine to implement a simple version for finding the best split that takes $O(dn^2)$ time.

See **regression_stumps.py** for starter code.

You need to complete the RegressionStump class by completing the following methods

- implement predict
- implement score
- implement fit

We advice to implement in the order specified.

7. BONUS exercise if time: Data that is not numbers

Question 1: Spam Filters
You are given the task to design a spam filter and you will be using **Linear Classification** and the perceptron algorithm (since that, and decision trees, is all we know yet).

The input data consists of a list of (email, spam/not spam label), and each email is represented by a variable length text string. Can you train a spam filter using this data using the perceptron algorithm and if so how? What issues do you see and do you have any ideas how they could be adressed?

Question 2: Categorical Features
You are solving a problem with machine learning and have decided to use linear classification (Perceptron). One of the data features is categorical and has four unordered values: Apple, Banana, Grape, Mango.

How could you use that feature in a linear classification setup? (The data should be a matrix of size $n \times d$ of real numbers.

8. BONUS exercise if time: Classification Stumps in O(n d lg n) time

In this exercise your job is to describe an algorithm that given a data set of labelled data (two classses only), constructs the binary classification tree (one internal node and two leafs) that minimize the 0-1 Loss over the training data. Such small classification trees are called classification stumps.

i.e. given data

$$D = \{(x_i, y_i) \mid 1 \leq i \leq n, y_i \in \{0, 1\}, x_i \in \mathbb{R}^d\}$$

construct the binary classification tree T that minimize

$$\frac{1}{n} \sum_{i=1}^{n} 1_{T(x_i) \neq y_i}$$

. Your algorithm must only use $O(nd \log n)$ time.

The root node considers only questions like $f_i < 42$ and this may be represented by the feature's index i and the value to compare with (42 here).

Hint: Consider each feature in turn and sort the data for that feature and permute the labels y with the same ordering and compute the score for each relevant split in $O(n \lg n)$ time