Assignment 1

Due to 17.11.2019

1.Submission Instructions

1.1 Theoretical Part

a) The objective of this part is to practice the terms and concepts in the supervised machine learning area.

b) If needed, make sure to explain or present an example to any given answer to assure full understanding of your idea.

c) Submission: please submit a PDF version named "ex1-theoretical-part.pdf".

1.2 Practical Part

a) The main objective of this part is to assure you know some basic and necessary methods in python to give a very basic and preliminary taste of a machine learning model.

b) You are allowed to only use standard python packages and numpy.

c) Use python3.7

d) Name your main code as ex1_main.py

Good Luck!

2. Questions

2.1 Theoretical Part

In this part you will compare between two strategies for the task of *online boolean conjunctions prediction*. The goal of this task is to predict the boolean conjunction that correctly represents the training set, where a *conjunction* is any product ($\wedge$) of literals (e.g. $\bar{x}_1 \wedge x_2$). The $\boldsymbol{H}$ = *hypothesis class* is defined as a set of all possible conjunctions over d variables (d the number of variables in the *boolean conjunction statement*).

A conjunction containing both versions of a literal (atomic and its negation) is interpreted as the *all-negative hypothesis*. Note that the hypothesis class includes the *all-negative hypothesis*

Example: given a training example=(0,1,1) and its labels=1, its predicted boolean conjunction could be: $\bar{x}_1 \wedge x_2 \wedge x_3$ (because obviously $\bar{0} \wedge 1 \wedge 1 = 1$). The hypothesis class(**H**) would be all of the possible conjunction of x_1, x_2 and x_3.

1. General definition of the problem

Recall that our goal is to predict the correct boolean conjunction.

1. Describe the instance domain (X) and the label set (Y).
2. Present the entire hypothesis class for d=2.
3. Suggest an equation that describes the size of the hypothesis class for any given d.
4. Suppose that the true conjunction is $\bar{x}_1 \wedge x_2 \wedge x_3$:
   (a) Can the following example exist in the training set: ((1,0,1,1),0)? If not, suggest a correction.
   (b) Can the following examples co-exist in the training set: ((0,1,1,0),1), ((0,1,1,1),1)? If not, suggest a correction.

2. The Consistency Algorithm

We will now present a prediction strategy to find the hypothesis that correctly represents the dataset. In general, in this approach we create an initial hypothesis, which will be updated in every iteration, so it would fit the dataset.

Given that the initial hypothesis is the *all-negative hypothesis*, this prediction strategy is described as follows:

Question 2:

1. Explain in your own word, what the Consistency Algorithm do?
2. Does this algorithm try to reduce the error on the training examples to zero? explain.
3. What is the run-time per iteration?

Algorithm 1 Consistency Algorithm

procedure THE CONSISTENCY ALGORITHM
  $h_0 = all\_negative\_hypothesis$
  for instance t in examples **do**
    $h^t = h^{t-1}$
    if $y^t = 1 \wedge \hat{y}^t = 0$ **then**
      for index i in instance t **do**
        if $x_i^t = 1$ **then**
          $remove\ \bar{x}_i\ from\ h^t$
        end if
        if $x_i^t = 0$ **then**
          $remove\ x_i\ from\ h^t$
        end if
      end for
    end if
    if $y^t = 0$ **then**
      $leave\ h^t\ as\ is$
    end if
  end for
end procedure

2.2 Practical Part:

In this part you will implement the *Consistency Algorithm* for the task of online boolean conjunction prediction. Given training examples and their classification, stored in a text file, and using the implemented algorithm you are expected to predict the boolean conjunction correctly.

1. Your main file should be called *boolean_conjunction_predictor.py*
2. The input text file path will be given as an input parameter (e.g. boolean_conjunction_predictor.py data.txt)
3. The following format will be used in the training file:
4. Each line represents a single example and classification, and is constructed by values 0,1 separated by white spaces. For example, say d = 3 then 0 1 1 1 is equivalent to $((0, 1, 1), 1) \in X \times Y$. Load the file and separate the training data into two containers:
   (a) An X matrix such that its columns correspond to the examples of the training data.
   (b) A Y vector such that its values correspond to the classification (the last column in the raw data).

   Loading the data from the files can be done using:

import numpy as np
training_examples = np.loadtxt(file_path)

5. implement the *Consistency Algorithm*

6. Your algorithm's final predicted boolean conjunction should be written to a file called *output.txt*. The following format should be used:

   In a single line separated by commas (',') you should write a list of the existing literals in the boolean conjunction - ordered by their index. Literals in the negation form should be written using the 'not(x_i)' notation.

   For example, say d=5 and the true boolean conjunction is: $x_1 \wedge \bar{x}_2 \wedge \bar{x}_4 \wedge x_5$ (notice that x3 is missing). Your algorithm should write to the file at the end of its run: **x1,not(x2),not(x4),x5**