ALBAN COBI

2.S986

DUE: 10 SEPT, 2020

PROBLEM SET #1

PROBLEM 1 NUMBER CLASSIFICATION DEMO

(a) WORKFLOW

① Acquire samples of handwritten numbers → ② Label numbers (manually) → ③ Standardize the data representation → ④ Find or develop correct classification algorithm → ⑤ Train algorithm → ⑥ Test/validate the algorithm

Example "Mechanical Turk"

1 2

(b) Time required for each step

① I imagine this can be done various ways. ~~Outsource~~ crowd this task to acquire data quickly or acquire documents of numbers and scan them to create a digital copy. By crowdsourcing it I'd estimate it can be done relatively quickly

my estimate: $\sim 10^2 - 10^3 \frac{\text{numbers}}{\text{hr.}}$

② By crowdsourcing you can also ask for labelling and acquire labels at the same rate as above. but to manually label I assume it takes

$$\frac{1 \text{ number}}{3 \text{ seconds}} * \frac{3600 \text{ sec}}{\text{hr}} = \frac{1200 \text{ numbers}}{\text{hr}}$$

③ Having digital (pixelated) copies of numbers and writing an algorithm to store the pixels in a vector or matrix will help standardize it. Developing the algorithm will take the largest. I would assume an expert

data scientist can do it in a day or two so all the numbers can be standardized in a few hours.

④ Again, I suspect an expert data scientist can pinpoint an algorithm quickly in order to classify each number, maybe on the same order as step ②.

⑤ Training the algorithm should not take long and probably on the same order of time as steps ③-④.

⑥ Testing and validating the algorithm should be quick, assuming its written. Probably on the order of magnitude less than steps ③-⑤.

$$\text{Time} \approx \text{Fixed Time} + \text{Variable Time}$$

$$\sim \underbrace{1 \text{ week for steps } 3\text{-}6}_{\text{Fixed Time}} + \underbrace{\frac{N_{TOTAL}}{\text{Rate of acquisition}} + \frac{N_{TOTAL}}{\text{Rate of labelling}}}_{\text{Variable Time}}$$

Most of the time is probably associated with setting up the automation Algorithm or procedure for each step.

PROBLEM 2 | PROBLEM 1 APPLIED TO MY RESEARCH

Ideas ① AI-BE experimental procedure

Performing experiments reacting aluminum fuel and water in the presence of another material to change the phase of that "other" material and generate high pressures

② Determine the importance/relation between material properties and geometrical properties of a suction cup and the resulting $\frac{\text{Force}}{\text{Area}} \triangleq$ Tenacity and Time of attachment

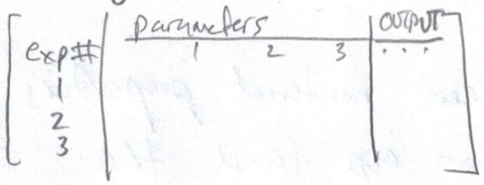
MASTER'S RESEARCH

I was fabricating soft suction cups with different material properties, in order to perform pull experiments to determine failure force. Ultimately I was aiming to determine scaling relationships between the force of failure and different material, geometrical, fluid properties.

For this exercise I will aim to determine what combination of parameters will achieve a pull force of a desired value

ML WORKFLOW

① Acquire experimental data that I already performed (already labeled)



② Standardize the data output such as using factorize() method from pandas library.

③ Find the correct classification algorithm

④ Train algorithm based on data I already have

⑤ Test/Validate the algorithm

I already have over 15-20 different experiments but I just have to put it into a table and organize it.

The step that should take me the longest is steps 2-3 because I don't have as much experience with determining the correct algorithm

I will take another stab at the workflow for the actual experiments I need to perform

EXPERIMENTAL WORKFLOW

Acquire materials needed

- silicone polymers
- newtonian fluids

↓

Decide what polymer to use, what fluid & what geometry for suction cups. to achieve target "detachment force"

→ Cast suction cup geometry

→ perform "detachment force" experiment & record maximum value.

↓

Determine error between actual force and target force.

← Decide whether to re-do experiment

← Decide which combinations of parameters to use (polymer, fluid, geometry)

↑

PROBLEM 3 MODIFY NUMBER CLASSIFICATION CODE

(a)

I chose the algorithm knn_classifier with hyperparameter n_neighbors=5 to achieve a classification accuracy of 99.44%.

(b) The drawbacks of tuning a "black box" machine learning algorithm are that I don't know how the algorithm works so it might be spitting out garbage. It could also mean that I'm potentially overfitting my data so that when I feed it new data, it will not perform as well.

Perhaps this is fine to try as a first test but without knowledge of how the algorithm works I would not be tempted to do it for my own research.