CS CAPSTONE PROGRESS REPORT

DECEMBER 7, 2019

REDUCING PATIENT DOSE FROM DIAGNOSTIC IMAGING USING MACHINE LEARNING

PREPARED FOR

OREGON STATE UNIVERSITY

DR. STEVE REESE

PREPARED BY

RADIOLOGICAL COUNTING

IAN BROWN

SEAN GILLEN

YIHONG LIU

Abstract

The capstone team has finished its preliminary documentation needed to start on the project and received approval from our client Dr. Reese. With sample code provided by Jessica Curtis and a data file provided by Dr. Reese, we are analyzing the data format with which we will use to train our machine learning models. We plan to spend some time before our meeting next term learning about the physical processes involved in spectroscopy, so that we can start using the Radiation Center's instruments to collect data. This document is covered under a Non-Disclosure Agreement (NDA) limiting access to its signers, the project's clients, and OSU employees (including Teaching Assistants).

Contents

1 Recap

The group will demonstrate a proof-of-concept reduction in spectroscopy classification time through the use of machine learning, following an algorithm developed collaboratively at OSU and Georgetown University. To do so, the team will use radiation counting data collected by an analogous detector set up at the Oregon State University Training, Research, Isotopes, General Atomics (TRIGA) reactor. Three different machine learning models will be trained and tested with this data to utilize their respective strengths.

2 Current Progress

The team has created a private GitHub repository, where we have adapted a script from OSU PhD candidate Jessica Curtis, who worked with Dr. Reese. This script reads in data from a file made by a detector in the Radiation Center, and records the relevant sections. We have researched machine learning technologies relevant to the algorithms we are using (neural network, naive Bayes, and decision tree). The basic architecture of the system is in our design document. From this, we can start the implementation of the main system components even though we cannot train the machine learning models yet.

3 Problems

Because the team has a CS-focused background, we have needed to spend time getting up to speed about how a spectrometer works. To remedy this, we are reading portions of the book *Radiation Detection and Measurement*, by Glenn Knoll, over winter break. The second main hurdle is that the data being used in the machine learning models is difficult to obtain because it must be taken from the lab in the Radiation Center. The problem of data should be resolved soon, because we plan to collect data at our next meeting.

4 Code

As part of our project, we have written a simple Python script that reads a file containing radiation counting data and creates a plot of it. It is adapted from a script provided to us by Jessica Curtis, at OSU. The code is included below.

```
import sys
import matplotlib.pylab as plt

data_filename = sys.argv[1]
data_filename_no_extension = data_filename.split('.')[0]

print('Program starting, filename is first arg:', data_filename)
print("raw file name:" + data_filename_no_extension)

outfileData = "output/" + data_filename_no_extension + "Data_1.csv"
outfileHeader = "output/" + data_filename_no_extension + "Header.csv"
outfileData = open(outfileData, 'w')
outfileHeader = open(outfileHeader, 'w')

in_header = True
initial_stage = True
detector_on = False
```

```
linecounter = 0
writeLine = ""

cycleID = 0
mS1counter = 0

dic_counts = {}

with open(data_filename, 'r') as fp:
    for line in fp:
        linecounter += 1

        if in_header:
            outfileHeader.write(line.replace('\r', ''))
            if 'Acq Start Reference' in line:
                in_header = 0
                outfileHeader.close()

        elif initial_stage and ('EX1' in line) and ('mS: 0' in line):
            initial_stage = False

        elif detector_on:
            # normal line write
            if 'ADC:' in line and 'Real:' in line and 'UTC Time:' in line:
                parse_string = line.split('Real:')[1]
                parse_string, time = parse_string.split('UTC Time:')
                parse_string, channel = parse_string.split('ADC:')
                milliSec = parse_string.split('mS')[0]

                writeLine = f'{cycleID},{milliSec},{channel}, {time.strip()}\n'
                channel = int(channel.strip())
                if channel in dic_counts:
                    dic_counts[channel] += 1
                else:
                    dic_counts[channel] = 1

                outfileData.write(writeLine)

            if ('EX1' in line) and ('mS: 0' in line):
                detector_on = False

        # create new file
        elif ('EX1' in line) and ('mS: 1' in line):
            detector_on = True

            mS1counter += 1
            cycleID += 1
            outfileData.close()
            outfileData = (
```

```
                f'output/{data_filename_no_extension}Data_{cycleID}.csv'
            )
            outfileData = open(outfileData, 'w')
            outfileData.write("runID, millisec, Bin, timestamp \n")
            print(
                f'lines read so far: {linecounter}, last line of '
                f'previous cycle: {writeLine}'
            )
            print('new cycle: {cycleID} line: {line}')

print(
    f'cycleCounter: {cycleID}, linecounter: {linecounter}, '
    f'mS1counter: {mS1counter}'
)

lists = sorted(dic_counts.items())
x, y = zip(*lists)
plt.plot(x, y)
plt.show()
# to see all millisec decimal places in R options(digits=10) and to load into R
# dataframe: filename<-read.csv("parsedFileName")
```

5 RETROSPECTIVE

Week	Positives	Deltas	Actions
1	None	None	None
2	Received starter code from Jessica	None	None
3	Signed the Non-disclosure Agreement	Understand the restrictions of the NDA	Meet with the client to discuss the NDA
4	Received more information on the NDA	Gain experience with Weka	Use included sample data with Weka to explore software features
5	Gained experience with Weka	Create an architecture diagram of our application	Use draw.io to create a UML diagram detailing our application architecture and share with the client
6	Toured the reactor bay at the Radiation Center on campus and received sample counting data	Use sample data file to create a plot of counts	Write Python code that reads the file and creates a plot using Matplotlib
7	Created a private GitHub repository	None	None
8	Implemented code quality fixes using linter	None	None
9	None	None	None
10	Submitted documents to Dr. Reese	Improve spectrometry knowledge	Read Ch. 3 and 12 of *Radiation Detection and Measurement*, by Knoll