Project 1: Extracting Time Series Properties of Glucose Levels in Artificial Pancreas

Introduction

Purpose

In this project, you will extract several performance metrics of an Artificial Pancreas system from sensor data.

Objectives

Learners will be able to:

- Extract feature data from a data set.
- Synchronize data from two sensors.
- Compute and report overall statistical measures from data.

Technology Requirements

- Python 3.6 to 3.8 (do not use 3.9).
- scikit-learn==0.21.2
- pandas==0.25.1
- Python pickle

Project Description

In this project, we are considering the Artificial Pancreas medical control system, specifically the Medtronic 670G system. The Medtronic system consists of a continuous glucose monitor (CGM) and the Guardian Sensor (12), which is used to collect blood glucose measurements every 5 minutes. The sensor is single-use and can be used continuously for 7 days after which it has to be replaced. The replacement procedures include a recalibratio requires the user to obtain blood glucose measurements using a Contour Next ®.

Note that this process also requires manual intervention. The Guardian Link Tr the CGM sensor and sends the data to the MiniMed 670G® insulin pump. The the Smart Guard Technology that modulates the insulin delivery based on the (SmartGuard Technology uses a Proportional, Integrative, and Derivative contrc bursts of insulin also called Micro bolus to be delivered to the user. During mea BolusWizard to compute the amount of food bolus required to maintain blood g user manually estimates the amount of carbohydrate intake and enters it to the

The Bolus Wizard is pre-configured with the correction factor, body weight, anc sensitivity of the subject, and it calculates the bolus insulin to be delivered. The program the MiniMed 670G infusion pump to deliver that amount. In addition tc MiniMed 670G insulin pump can also provide a correction bolus. The correctio provided only if the CGM reading is above a threshold (typically 120 mg/dL) ar amount with respect to the difference of the CGM reading and the threshold.

The SmartGuard technology has two methods of suspending insulin delivery: a where the insulin delivery is stopped when the CGM reading is less than a cert suspend on predicted low, where the insulin delivery is stopped when the CGM to be less than a certain threshold. Apart from these options, insulin delivery ca manually by the user or can be suspended when the insulin reservoir is runnin

Directions

Dataset:

You will be given two datasets:

1. From the Continuous Glucose Sensor (CGMData.csv) and
2. from the insulin pump (InsulinData.csv)

The output of the CGM sensor consists of three columns:

1. **Data time stamp (Columns B and C combined)**,
2. **the 5 minute filtered CGM reading in mg/dL**, (Column AE) and
3. the ISIG value which is the raw sensor output every 5 mins.

The output of the pump has the following information:

1. **Data time stamp**,

2. Basal setting,
3. Micro bolus every 5 mins,
4. Meal intake amount in terms of grams of carbohydrate,
5. Meal bolus,
6. correction bolus,
7. correction factor,
8. CGM calibration or insulin reservoir-related alarms, and
9. **auto mode exit events and unique codes representing reasons (Cc**

The bold items are the columns that you will be using in this assignment.

Metrics to be extracted:

1. Percentage time in hyperglycemia (CGM > 180 mg/dL),
2. percentage of time in hyperglycemia critical (CGM > 250 mg/dL),
3. percentage time in range (CGM >= 70 mg/dL and CGM <= 180 mg/dL),
4. percentage time in range secondary (CGM >= 70 mg/dL and CGM <= 1
5. percentage time in hypoglycemia level 1 (CGM < 70 mg/dL), and
6. percentage time in hypoglycemia level 2 (CGM < 54 mg/dL).

Each of the above-mentioned metrics are extracted in three different time inter midnight), overnight (midnight to 6 am), and whole day (12 am to 12 am).

Percentage is with respect to the total number of CGM data that should be ava Assume that the total number of CGM data that should be available is 288. Th that the number of data available is less than 288, but still consider the percen to 288.

You have to extract these metrics for each day and then report the mean value all days. Hence there are 18 metrics to be extracted.

The metrics will be computed for two cases:

- Case A: Manual mode
- Case B: Auto mode

Analysis Procedure:

The data is in reverse order of time. This means that the first row is the end of
whereas the last row is the beginning of the data collection. The data starts wit
Manual mode continues until you get a message “AUTO MODE ACTIVE PLGM
“Q” of the InsulinData.csv. From then onwards Auto mode starts. *You may get
ACTIVE PLGM OFF” in column “Q” but only use the earliest one to determine
auto mode. There is no switching back to manual mode,* so the first task is to c
stamp when Auto mode starts. **Remember that the time stamp of the CGM c
as the timestamp of the insulin pump data because these are two differer
operate asynchronously.**

**Once you determine the start of Auto Mode from InsulinData.csv, you hav
timestamp in CGMData.csv where Auto mode starts. This can be done sir
for the time stamp that is nearest to (and later than) the Auto mode start t
from InsulinData.csv.**

For each user, CGM data is first parsed and divided into segments, where eac
corresponds to a day worth of data. One day is considered to start at 12 am ar
there is no CGM data loss, then there should be 288 samples in each segmen
whole is used to compute the metrics for the whole day time period. Each segr
into two sub-segments: daytime sub-segment and overnight subsegment. For
CGM series is investigated to count the number of samples that belong to the
metrics. To compute the percentage with respect to 24 hours, the total number
specified range is divided by 288.

Note that here you have to tackle the “missing data problem”, so a particular m
data points. In the data files, those are represented as NaN. You need to devis
the missing data problem. Popular strategies include deletion of the entire day
interpolation.

Write a Python script that accepts two CSV files: CGMData.csv and InsulinDat
analysis procedure and outputs the metrics discussed in the metrics section in
the format described in Results.csv.

Submission Directions for Project Delive

Submit your Python script and Results.csv file ir Canvas.

Deliverables:

- Code (‘main.py.’ code file)
- Results.csv
  - Do NOT include headers
  - Your Results.csv should only contain numbers and it should be a 2 X 18
- Requirements file (detailing how the requirements were fulfilled). See attac
  template for an example. Save the Requirements file as a "Requirements.t

Submission Guidelines:

- Please submit a zipped file containing 1, 2, and 3 deliverables as “**yourfirstname_lastname_Project1.zip**".
  - Do not create an additional folder; just zip the files directly.
- **The submission space is located at the bottom of module 2 Assignment: Project 1: Extracting Time Series Properties of in Artificial Pancreas Submission”.**

Template_Requirements-1.docx (https://canvas.asu.edu/courses/140907/files/

Project-1-Files-1.zip (https://canvas.asu.edu/courses/140907/files/60728989?wr
(https://canvas.asu.edu/courses/140907/files/60728989/download?download_frd=