MCS Portfolio Project Report

Maclay Teefey
CSE 572: Data Mining

Arizona State University
Tempe, Arizona, United States
mjteefey@asu.edu

***Abstract*—The processing and analysis of sensor data has been an essential piece of managing the health of diabetes patients. This report's goal is to describe the process and results of extracting and analyzing sensor data from an artificial pancreas system. The project report describes the creation of a machine learning model to predict whether a person ate a meal or didn't eat a meal in a sequence of sensor data. Two sets of data from a glucose monitor and an insulin pump are split into sequences that a person ate a meal or didn't eat a meal. Various features are created to distinguish the meal and no meal data from each other. A Random Forest Model is trained through the data fed through a pipeline of the created features and correctly predicted 57.6% of unlabeled test data. This project report contains a description of the creation of a clustering model that accurately clusters meal data into groups based on the amount of carbohydrates eaten for the patient's meal. Training sensor data calculated six ground truth bins before removing the carbohydrate information from the training data. Two clustering techniques, K-Means and Density-based spatial clustering of applications with noise (DBSCAN), were used to group the unlabeled training data using features created in the first project. The clusters created by the two clustering techniques were evaluated using the ground truth bins for the sum of squared errors, purity, and entropy. Both clustering methods had an entropy of 0.0 and a purity of 1.0, but the K-Means clusters have an SSE of 100 and the DBSCAN clusters had an SSE of 305.**

Keywords— Data Mining, Machine Learning, Clustering, Data Analytics, Data Science

I. Introduction

Wearable sensors have become a tool for around one third of Americans to track their health [1]. For adults with diabetes 1, this rate is even higher at above 48% due to the widespread adoption of Continuous Glucose Monitors (CGMs) [2]. The report's goal is to report on data mining techniques for a system that utilizes CGMs called the Medtronic 670G artificial pancreas medical control system. The Medtronic system uses two sensors: a glucose sensor, and a MiniMed 670G insulin pump. To deliver insulin to the patient, the insulin pump calculates how much insulin is required to maintain the patient's insulin level from the glucose sensor. When the CGM reading is below a certain threshold or is predicted to be less than a certain threshold, the insulin pump stops delivering insulin to the patient. The CGM sensor reports the date and time of the sensor, the glucose value (mg/dL), ISIG value, and any sensor errors. The insulin pump sensor reports the date time of the sensor, basal setting, micro bolus every 5 mins, meal intake in terms of grams of carbohydrates, meal bolus, correction bolus, correction bolus, correction factor, and other information (CGM calibration, insulin alarms, auto mode exit events, and unique codes). Both sensors record and report the sensor's information every five minutes, so there are 288 segments across a 24-hour period. The CGM sensor data used spans from 12:08 PM on November 25th, 2017, to 1:22 PM on February 12th, 2018. Insulin sensor data spans from 6:59 PM on November 24th, 2017, to 1:20 PM on February 12th, 2018. The two sensors are not synchronized to each other, and both sensors have missing segments and segments with missing values. The meal intake amount is manually input into the insulin sensor, but this can result into problems gaging the that should be provided to the patient. For example, the patient may report their meal amount after eating a meal, which can leave a large time gap between when they first started eating their meal and finish eating the meal to provide insulin. There can also be inaccuracies with how many carbohydrates are eaten with each meal. To help mitigate the meal input concerns, this report describes a machine learning model to predict whether a patient ate a meal and two clustering models to group meal intake amounts without directly referencing the meal intake amount.

II. Explanation of Solutions

A. Labeling Meal Data

To distinguish between meal data and no meal data, the sensor data must extract training meal and no meal data sets. To accomplish this, sensor data where a carb intake was recorded will be considered for extraction if there were no meals recorded from the carb intake report to two hours later. Once that condition is met, the previous 30 minutes to the meal intake and the 2 hours following the meal intake are recorded as a sequence of meal data. After a meal is recorded, two hour sequences after the sequence of meal data are recorded as no meal data if they do not have any carb intake reported in the 2-hour sequence.

B. Handling Missing Data

If only complete sequence data were utilized for this project, there would not be enough training data to proceed. To be able to have enough data points to train both machine learning models, a threshold of 20% missing data point was set. For the sequences that include 80% or more of the data points, the missing glucose level values were set using linear interpolation.

C. Meal Data Distribution

When a person eats a meal, their blood glucose curve increases logarithmically until 20 minutes after it reaches its peak. The glucose level decreases from the peak over the next two hours until it returns around the person's resting glucose level [3, Fig 1]. The prominence of the peak and the amount of time it takes to return to the resting glucose level is dependent on the amount of carbohydrates consumed. If the person does not eat a meal, their glucose level will vary across a period due to natural variations in a person's glucose level and does not represent a

trend [4]. To distinguish between meal and no meal data and the amount of carbohydrates eaten in a meal, various features can be created to reflect the data distribution issues.

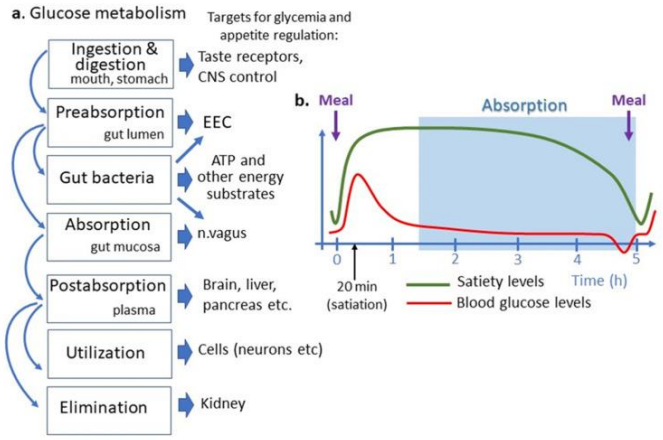

Fig. 1. Glucose metabolism process between meal times

D. Meal Data Factors

The first category of features used are basic statistical features: mean glucose level, median glucose level, standard deviation of glucose level, variance of the glucose level, minimum glucose level, maximum glucose level, the range of glucose levels, and the skew and kurtosis of the data distribution.

The second category of features used are derivative features. Due to the glucose level resembling a sine wave after eating a meal, the first and second derivatives will show a pattern resembling the velocity and acceleration curves of a sine wave if a person eats a meal. If the person did not eat a meal, the first and second derivative charts should not show a pattern and should reflect the natural variations in glucose levels across a day. For the first derivative, the ratio of increasing glucose periods, the average velocity, the standard deviation of the velocity, the maximum increase in velocity, the maximum decrease in velocity, and the average magnitude of velocity were used. For the second derivative, the ratio of accelerating glucose periods, the mean acceleration, the standard deviation of the acceleration, max acceleration, and min acceleration were utilized. Along with the derivative, the area under the curve and max consecutive increase in area under the curve were utilized as a feature.

The third category of features focuses on peak detection. The meal data should have a much more prominent peak than any other peak in the data sequence and should reach its peak around the same time as other meal sequences. Using SciPy's find peaks function, the number of peaks, the max peak height, the average peak height, the timing of the first peak, the average peak prominence, and average peak width are used as features.

The fourth category of features focuses on sine and frequency analysis. If the data is treated like a sine wave, the data should have closer frequencies and power to each other than no data. The dominant frequency, amplitude, power, and fit quality of the calculated sine wave are used as features. Using SciPy's Fast Fourier Transformation (FFT) function, the dominant frequency, power, total spectral power, spectral centroid, and spectral rolloff are extracted from the data to be utilized as features.

The final category of features tries to split the data into various sections. One set of features splits the data into 30-minute sections to better delineate the meal data distribution and non-meal data distribution. This feature extraction strategy splits the data into four 30-minute bins from 0 to 120 minutes. For each bin, the mean, median, range, max, min, and slope are calculated with all of them being individual features. Another split utilized is through splitting the data into the first half and second half. The change in slope between the first and second half, and the ratio between area under the curve for the second half and the first half were utilized as features.

E. Machine Learning Model

For the classification of meal and no meal data, the training data is run through the feature extraction pipeline and scaled using scikit-learn's StandardScalar. The model utilized for this project is scikit-learn's RandomForestClassifier with cross validation to prevent overfitting.

F. Clustering Models

For the clustering of meal data based on carbohydrate intake amounts, two models were used: K-Means and DBSCAN clustering. The meal data is split into ground truth bins of size 20 from the minimum carbohydrate intake to maximum carbohydrate intake. After calculating the ground truth bins, the truth labels were removed from the training data and the clustering models were applied. For K-Means, the model will use the best clusters from ski-learn's KMeans function using 50 different starting seeds and 1000 iterations to find each final clustering. For DBSCAN, the model uses the best clusters from ski-learn's DBSCAN function. To help DBSCAN select the correct clusters, combinations of k-distance percentiles and eps values are selected. If the number of generated density-based clusters does not match the number of ground truth bins, the highest error clusters are split using bisecting k-means until there are as many DBSCAN clusters as ground truth bins. Because the K-Means and DBSCAN clustering models are unsupervised models, the verification of clusters is more complicated than verifying the results of the supervised models. To verify that the correct ground truth bins match up with the best fitting generated cluster, Hungarian assignment is assigned to both clustering models. Once the generated clusters match the correct ground truth bins, the models can be evaluated.

III. Description of Results

A. Machine Learning Model

To verify the predictive ability of the machine learning model, the model runs a test data set and reports the metrics accuracy and F1 score. The machine learning model had an overall accuracy of 57.58% and an F1 score of 0.57.

B. Clustering Models

For the K-Means and DBSCAN models, both unsupervised and supervised evaluation metrics are utilized. SSE calculates the distance from each cluster data point to the cluster center and sums across all clusters. The K-Means model clusters have an SSE of 100.04, and DBSCAN model clusters have an SSE of 305.03. Because the ground truth bins were calculated prior to feature extraction, the purity and entropy can be calculated for

these clusters. For both the K-Means and DBSCAN models, the clusters had an entropy of 0 and purity of 1. This means that every meal data sequence was properly grouped into the correct cluster using both models.

IV. Description of Contributions

The two projects were completed individually, so this report and the code for the two projects were entirely done by me.

V. Self Reflection

To accomplish the creation of machine learning and clustering models for sensor data, multiple new skills were learned.

A. Data Processing

When analyzing data from the insulin sensor data and the CGM sensor data, issues would occur due to the insulin sensor being offset from each other by three minutes. Before this project, I had not analyzed more than one sensor csv and have not had to join two data sources more complicated than creating primary keys, so this created a good challenge.

B. Feature Extraction

For most of my previous work creating machine learning models, I have not been tasked with the creation of all the features of the model. This project forced me to learn how make multiple different categories of features to fully capture a data distribution. Due to the data distribution resembling a sine wave, I applied two different methods of finding the frequency of the data: sine wave fitting and Fourier frequency transformation.

C. Clustering

The technique of clustering through the density of points in a distribution was completely new to me prior to taking CSE 572, so I was not exposed to the necessary steps to use DBSCAN. With the clustering of DBSCAN, I have never needed to manually split my clusters through bisecting k-means, because I have not been required to force a model to extract an extra couple clusters. I have also not had to use hungarian assignment before, but it was necessary to verify and make correct the ground true bins assignment.

References

[1] *L. S. Dhingra et al., "Use of Wearable Devices in Individuals With or at Risk for Cardiovascular Disease in the US, 2019 to 2020," JAMA Netw Open, vol. 6, no. 6, p. e2316634, Jun. 2023, doi: 10.1001/jamanetworkopen.2023.16634.*

[2] D. J. DeSalvo *et al.*, "Patient Demographics and Clinical Outcomes Among Type 1 Diabetes Patients Using Continuous Glucose Monitors: Data From T1D Exchange Real-World Observational Study," *J Diabetes Sci Technol*, vol. 17, no. 2, pp. 322–328, Mar. 2023, doi: 10.1177/19322968211049783.

[3] L. V. Gromova, S. O. Fetissov, and A. A. Gruzdkov, "Mechanisms of Glucose Absorption in the Small Intestine in Health and Metabolic Diseases and Their Role in Appetite Regulation," *Nutrients*, vol. 13, no. 7, p. 2474, Jul. 2021, doi: 10.3390/nu13072474.

[4] S. Moebus, L. Göres, C. Lösch, and K.-H. Jöckel, "Impact of time since last caloric intake on blood glucose levels," *Eur J Epidemiol*, vol. 26, no. 9, pp. 719–728, Sep. 2011, doi: 10.1007/s10654-011-9608-z.