UNIVERSITY OF BEIRA INTERIOR

Personalized Prediction of Hypoglycemia Events

João Tomás Mota Cunha

1st Cycle in Biomedical Sciences
2020/20201

Tutors
Prof. Doutor Nuno Garcia
Prof. Doutor Nuno Pombo
Mestre Virginie Felizardo

Covilhã, june de 2021

Abstract

New artificial intelligence technologies are increasingly present in the health area. An example of this is the use of machine learning techniques to predict hypoglycemia events. This work presents a proposal for personalized prediction, performed in 5 patients of hypoglycemia events using the classifiers: AdaBoost Tree, Bagged Tree, and Subspace KNN. Previously selected and adapted training and test datasets were used. Each classifier has a set of parameters adjusted to present a better prediction for each patient. Based on the results obtained, it was possible to calculate the number of “false alarms” and the percentage of predicted hypoglycemia. We found that the most effective classifier was the Bagged Tree, due to its low “false alarms” values, being able to predict most of the hypoglycemia events. The worst results are probably due to the inadequate choice of values for each parameter, influencing the results obtained. With the application of these methods, we conclude that the use of machine learning techniques is very important in the role played by biomedical engineers, making the improvement of these applications more accessible, enabling increasingly accurate and effective predictions, to improve the quality of life of diabetics.

Introduction

Diabetes mellitus (DM) is a chronic disease characterized by the inability to regulate blood glucose (BG) normally, primarily due to the body's failure to produce insulin (Type 1 diabetes) or the body's inability to respond to insulin's action (Type 2 diabetes), as well as gestational diabetes[1]. According to the International Diabetes Federation (IDF), in 2015, there were 415 million people with diabetes worldwide, making it one of the greatest global health emergencies of the 21st century[2]. People with diabetes have higher rates of morbidity and mortality compared to the general population[3].

Hypoglycemia, or a drop in BG below critical levels, is defined as the decrease in BG below 70 mg/dL or 3.9 mmol/L[4]. It is one of the most common and lethal conditions for both Type 1 and Type 2 diabetics[5]. Hypoglycemia can cause symptoms such as loss of consciousness, confusion, seizures, and in extreme cases, death. However, these symptoms can vary among individuals depending on various factors[6]. Moreover, based on BG levels, hypoglycemia symptoms can be classified as mild, moderate, or severe. In most cases, hypoglycemia occurs silently without any symptoms[5].

Over time, there has been a significant increase in the use of technology to manage diabetes, particularly with glucose monitoring systems, which have become a trending topic in biomedicine[7]. For this reason, biomedical engineers aim to create increasingly efficient prediction models to reduce uncertainty and improve the quality of life for diabetics.

The complexity of diabetes prognosis and management has led to the integration of artificial intelligence (AI) and machine learning (ML) techniques as key technologies for diabetes treatment[3]. This explains the exponential growth in research focused on ML techniques for predicting adverse glycemic events in general and hypoglycemia in particular[8].

ML has recently become one of the main fields of AI, and its impact on healthcare has been enormous[9][10]. The concept of ML is rooted in computer science, statistics, and optimization, with the goal of enabling computers to train themselves without being explicitly programmed, giving them the ability to predict outcomes with a certain level of accuracy. In many medical scenarios, prior knowledge of an adverse event can prevent an emergency and, in many cases, save lives. The ability of ML to predict future events thus becomes a powerful tool to anticipate such occurrences[11].

Currently, various algorithms are used to predict diabetes, including traditional ML methods. For instance, in the work of Kavakiotis et al. (2017)[12], they conducted a systematic review of ML and data mining techniques using different classifiers to study diabetes in the context of prediction and diagnosis, diabetes complications, genetic history and environment, and

healthcare. Another example is the work of Razavian et al. (2015)[13], who, in order to handle high-dimensional datasets, built predictive models based on logistic regression for different onset predictions of Type 2 diabetes. Similarly, Georga et al. (2013)[14] focused on blood glucose and used support vector regression (SVR) to predict diabetes, a multivariate regression problem.

In this project, the goal is to create a personalized prediction of hypoglycemia events using the Python programming language, by employing the classifiers: AdaBoost Tree, Bagged Tree, and Subspace KNN. To achieve personalized predictions, the parameters of each classifier were adjusted according to the patient's data.

The rest of the report is organized as follows: "Materials and Methods," where concepts of Python, machine learning, classifiers, and the dataset used are discussed; "Results and Discussion," where the results obtained are evaluated and discussed; and finally, the "Conclusion," where the findings of the study are presented.

Materials and Methods

1) Python and Machine Learning

Python is an example of a programming language widely used in Machine Learning (ML). It has proven to be highly effective and is utilized in many scientific computing applications[15]. By using pre-packaged algorithms, it is possible to apply Python in ML processes, which are used to solve large-scale tasks at a slower speed and require significant memory and computational capacity[16].

ML is a type of Artificial Intelligence (AI) that encompasses algorithmic methods allowing models to "learn" without being explicitly programmed. In other words, ML algorithms enable models to learn on their own once the learning algorithm is determined. This makes the model more adaptable to the uncertain and ever-changing real-world environment. ML addresses the challenge of how to build computational programs that automatically improve through experience[17].

2) Classifiers

***Ada-boost*:** One of the ensemble boosting classifiers that combines multiple classifiers to enhance accuracy. Through an iterative ensemble method, AdaBoost builds a strong classifier by combining several weak classifiers, resulting in a highly accurate strong classifier. The basic

concept behind AdaBoost is to assign weights to the classifiers and train the data sample in each iteration to ensure accurate predictions of unusual observations[18].

Bagged Tree: Using ensemble methods that combine several decision trees, improves predictive performance compared to using a single decision tree. It is an ensemble technique that creates a collection of predictors. Consecutive trees (random samples) are fitted, and at each step, the goal is to resolve the net error of the previous tree[19].

***Subspace KNN*:** This is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption about the underlying data distribution, meaning the model structure is determined by the dataset. A lazy algorithm means that no training data is needed for model generation; all training data is used during the testing phase. This makes training faster, but testing slower and more resource-intensive, as it requires more memory to store the training data. In the worst case, KNN needs more time to check all data points, and verifying all data points will require more memory to store the training data[20].

3) Datasets

The framework design was executed using a diabetes dataset[21] from the UCI Machine Learning Repository, which contains data from 70 Type 1 diabetic patients. The records of diabetic patients from the UCI dataset were obtained from two sources: an automatic electronic recording device with an internal clock for event logging and paper records that provided only time intervals related to the time of day, such as breakfast, lunch, dinner, and bedtime.

The provided datasets had already undergone preprocessing, including data manipulation, variable extraction, blood glucose imputation (in cases where data was missing), and resampling to obtain a balanced dataset.

4) Proposed Model

The proposed model for personalized hypoglycemia event prediction involves the use of three machine learning classifiers, a type of artificial intelligence: AdaBoost Tree, Bagged Tree, and Subspace KNN.

Note: The characteristic computational code for the AdaBoost Tree classifier is depicted in Annex 1, using the Spyder editor and the Python programming language. The only part of the code that differs between classifiers is the "model construction," which appears between lines 26 and 30. This section of the code needs to be adjusted because it is responsible for choosing the parameters used for each patient.

This personalized prediction will be performed for patients 55, 56, 65, 67, and 68. For each patient, we created a training dataset (data from 52 patients + data with 80% of the patient's

hypoglycemia episodes) and a test dataset (data with 20% of the patient's hypoglycemia episodes). These datasets are described in Table 1.

Each classifier has a set of parameters, and these were adjusted to provide the best prediction for each patient, as shown in Table 2.

Note: Depending on the classifier, parameters such as the number of estimators (AdaBoost Tree), number of trees and seed (Bagged Tree), and max features (Subspace KNN) were modified.

Table 1: Dataset Description.

Patients	**Training Dataset** Number of Hypoglycemia Episodes (Total Observations)	**Test Dataset** Number of Hypoglycemia Episodes (Total Observations)
P55 (27 hypoglycemias)	21 (26349)	6 (76)
P56 (72 hypoglycemias)	57 (26249)	15 (68)
P65 (20 hypoglycemias)	16 (26289)	4 (134)
P67 (24 hypoglycemias)	19 (26217)	5 (184)
P68 (19 hypoglycemias)	15 (26090)	4 (291)

Table 2: Description of Parameters Used.

Classifier	Parameter 1	Parameter 2	Parameter 3
AdaBoost Tree	N° estimators = ..	Learning rate = 1	Algorithm = 'SAMME'
Bagged Tree	Base estimator = cart	N° estimators = n° trees	Random state = seed
Subspace KNN	Base estimator = KNeighborsClassifier(2)	Bootstrap = False	Max features = ..

Results

Table 3 presents the results obtained using the AdaBoost classifier for five patients (P55, P56, P65, P67, and P68). The "number of estimators" was individually adjusted for each patient, with values ranging from 20 to 1750. The analysis focused on the number of "false alarms" and the percentage of predicted hypoglycemic events.

Table 3: Results of the AdaBoost Classifier.

N° estimators = 20	P55		
[63 0 0]		precision	recall
[5 1 0]	0	0.93	1.00
[0 0 6]	1	1.00	0.17
Accuracy: 0.93	2	1.00	1.00
N° estimators = 630	P56		
[11 8 0]		precision	recall
[9 24 0]	0	0.55	0.58
[0 0 15]	1	0.75	0.73
Accuracy: 0.75	2	1.00	1.00
N° estimators = 105	P65		
[68 48 0]		precision	recall
[5 8 0]	0	0.93	0.59
[0 0 4]	1	0.14	0.62
Accuracy: 0.60	2	1.00	1.00

Nº estimators = 1750	P67		
[123 34 0]		precision	recall
[14 7 0]	0	0.90	0.78
[0 0 5]	1	0.17	0.33
Accuracy: 0.74	2	1.00	1.00
Nº estimators = 80	P68		
[135 117 0]		precision	recall
[15 19 0]	0	0.90	0.54
[0 0 4]	1	0.14	0.56
Accuracy: 0.54	2	1.00	1.00

AdaBoost Tree		
Patient	False Alarms	Hypoglycemia Predicted %
P55	0	0,75
P56	0,235294118	0,93
P65	0,4	1
P67	0,209876543	1
P68	0,45703125	1

The results show that while three patients (P55, P56, P67) had "false alarms" values below 30%, the AdaBoost classifier demonstrated relatively low precision, especially for patients P68 and P65. Hypoglycemia predictions were mostly successful, with high percentages across the patients, but the variability in performance suggests inconsistent precision.

Table 4: Results of the Bagged Tree Classifier

Nº trees = 90	seed = 8	P55	
[55 8 0]		precision	recall
[2 4 0]	0	0.96	0.87
[0 0 6]	1	0.33	0.67
Accuracy: 0.87	2	1.00	1.00
Nº trees = 30	seed = 7	P56	
[14 5 0]		precision	recall
[24 9 0]	0	0.37	0.74
[0 0 15]	1	0.64	0.27
Accuracy: 0.57	2	1.00	1.00
Nº trees = 20	seed = 7	P65	
[104 12 0]		precision	recall
[8 5 0]	0	0.93	0.90
[0 0 4]	1	0.29	0.38
Accuracy: 0.85	2	1.00	1.00
Nº trees = 100	seed = 7	P67	
[139 18 0]		precision	recall
[17 4 0]	0	0.89	0.89
[0 0 5]	1	0.18	0.19
Accuracy: 0.81	2	1.00	1.00
Nº trees = 50	seed = 20	P68	
[230 22 0]		precision	recall
[23 11 0]	0	0.91	0.91
[0 0 4]	1	0.33	0.32
Accuracy: 0.84	2	1.00	1.00

Bagged Tree		
Patient	False Alarms	Hypoglycemia Predicted %
P55	0,115942029	1
P56	0,147058824	0,4
P65	0,1	1
P67	0,111111111	0,4
P68	0,086614173	1

The Bagged Tree classifier outperformed AdaBoost, with all patients showing false alarms below 30%, indicating a higher overall precision. Hypoglycemia predictions were efficient for most patients, except for P56 and P67, where only 40% of events were predicted.

Table 5: Results of the Subspace KNN Classifier

Max features = 12	P55		
[46 17 0]		precision	recall
[2 4 0]	0	0.96	0.73
[0 0 6]	1	0.19	0.67
Accuracy: 0.75	2	1.00	1.00
Max features = 5	P56		
[10 9 0]		precision	recall
[11 17 5]	0	0.48	0.53
[0 0 15]	1	0.65	0.52
Accuracy: 0.63	2	0.75	1.00
Max features = 7	P65		
[60 56 0]		precision	recall
[7 6 0]	0	0.90	0.52
[0 0 4]	1	0.10	0.46
Accuracy: 0.53	2	1.00	1.00
Max features = 4	P67		
[99 52 6]		precision	recall
[13 8 0]	0	0.88	0.63
[0 0 5]	1	0.13	0.38
Accuracy: 0.61	2	0.45	1.00
Max features = 3	P68		
[154 93 5]		precision	recall
[16 16 2]	0	0.91	0.61
[0 0 4]	1	0.15	0.47
Accuracy: 0.60	2	0.36	1.00

Subspace KNN		
Patient	False Alarms	Hypoglycemia Predicted %
P55	0,246376812	1
P56	0,264705882	0,93
P65	0,466666667	1
P67	0,333333333	1
P68	0,370517928	1

The Subspace KNN classifier performed worse compared to the Bagged Tree, with only two patients (P55 and P56) showing false alarms below 30%. While hypoglycemia predictions were

reasonable, the overall precision of the classifier was lower, particularly for patients P65, P67, and P68.

Discussion

From the three tables presented, the Bagged Tree classifier emerged as the most effective, showing lower false alarms and more robust predictive performance. On the other hand, the Subspace KNN classifier had significantly lower accuracy, and the AdaBoost classifier exhibited performance variability between patients.

The difficulties encountered by the AdaBoost and Subspace KNN classifiers may be attributed to the sensitivity of parameter selection, which varies from patient to patient. The process of adjusting these parameters can be time-consuming and complex, potentially contributing to less favorable results.

Overall, the study highlights the importance of tailoring classifier parameters to each patient, as well as selecting the most appropriate model to predict hypoglycemic events accurately.

Conclusion

This study aimed to provide personalized predictions of hypoglycemic events using three machine learning classifiers (AdaBoost Tree, Bagged Tree, and Subspace KNN) for five type 1 diabetic patients. The Bagged Tree classifier proved to be the most effective, with low false alarm rates and high accuracy, while the AdaBoost and Subspace KNN classifiers struggled, largely due to parameter selection challenges.

These results underscore the importance of fine-tuning classifier parameters for each individual, which adds complexity to the process and requires considerable time. The study contributes to the development of more precise and personalized tools for predicting hypoglycemic events, which can help improve glycemic control for diabetic patients and enhance their quality of life.

Furthermore, the research emphasizes the need for continued improvements and refinements in machine learning models applied to hypoglycemia prediction, focusing on both personalization and predictive efficiency.

References

[1] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting Diabetes Mellitus With Machine Learning Techniques," *Front. Genet.*, vol. 9, p. 515, Nov. 2018, doi: 10.3389/fgene.2018.00515.

[2] S. Oviedo, J. Vehí, R. Calm, and J. Armengol, "A review of personalized blood glucose prediction strategies for T1DM patients," *Int. j. numer. method. biomed. eng.*, vol. 33, no. 6, pp. 1–21, 2017, doi: 10.1002/cnm.2833.

[3] A. Z. Woldaregay *et al.*, "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes," *Artif. Intell. Med.*, vol. 98, no. August 2018, pp. 109–134, 2019, doi: 10.1016/j.artmed.2019.07.007.

[4] J. F. Yale, B. Paty, and P. A. Senior, "Hypoglycemia," *Can. J. Diabetes*, vol. 42, pp. S104–S108, Apr. 2018, doi: 10.1016/j.jcjd.2017.10.010.

[5] O. Mujahid, I. Contreras, and J. Vehi, "Machine learning techniques for hypoglycemia prediction: Trends and challenges," *Sensors (Switzerland)*, vol. 21, no. 2, pp. 1–21, 2021, doi: 10.3390/s21020546.

[6] "IDF Diabetes Atlas." https://www.idf.org/e-library/epidemiology-research/diabetes-atlas/159-idf-diabetes-atlas-ninth-edition-2019.html (accessed Jun. 04, 2021).

[7] P. Shende, P. Sahu, and R. Gaud, "A technology roadmap of smart biosensors from conventional glucose monitoring systems," *Therapeutic Delivery*, vol. 8, no. 6. Future Medicine Ltd., pp. 411–423, Jun. 01, 2017, doi: 10.4155/tde-2017-0012.

[8] I. Contreras and J. Vehi, "Artificial intelligence for diabetes management and decision support: Literature review," *Journal of Medical Internet Research*, vol. 20, no. 5. JMIR Publications Inc., p. e10775, May 30, 2018, doi: 10.2196/10775.

[9] J. Wiens and E. S. Shenoy, "Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology," *Clin. Infect. Dis.*, vol. 66, no. 1, pp. 149–153, Jan. 2018, doi: 10.1093/cid/cix731.

[10] J. A. Roth, M. Battegay, F. Juchler, J. E. Vogt, and A. F. Widmer, "Introduction to Machine Learning in Digital Healthcare Epidemiology," *Infect. Control Hosp. Epidemiol.*, vol. 39, no. 12, pp. 1457–1462, Dec. 2018, doi: 10.1017/ice.2018.265.

[11] R. C. Deo, "Machine learning in medicine," *Circulation*, vol. 132, no. 20, pp. 1920–

1930, Nov. 2015, doi: 10.1161/CIRCULATIONAHA.115.001593.

[12] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, “Machine Learning and Data Mining Methods in Diabetes Research,” *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 104–116, 2017, doi: 10.1016/j.csbj.2016.12.005.

[13] N. Razavian, S. Blecker, A. M. Schmidt, A. Smith-Mclallen, S. Nigam, and D. Sontag, “Population-level prediction of type 2 diabetes from claims data and analysis of risk factors,” *Big Data*, vol. 3, no. 4, pp. 277–287, 2015, doi: 10.1089/big.2015.0020.

[14] E. I. Georga *et al.*, “Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression,” *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 1, pp. 71–81, 2013, doi: 10.1109/TITB.2012.2219876.

[15] E. Serrano, J. G. Blas, J. Carretero, M. Abella, and M. Desco, “Medical Imaging Processing on a Big Data Platform Using Python: Experiences with Heterogeneous and Homogeneous Architectures,” in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 830–837, doi: 10.1109/CCGRID.2017.56.

[16] K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, and Z. Huo, “Julia language in machine learning: Algorithms, applications, and open issues,” *Computer Science Review*, vol. 37. Elsevier Ireland Ltd, p. 100254, Aug. 01, 2020, doi: 10.1016/j.cosrev.2020.100254.

[17] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science (80-.).*, vol. 349, no. 6245, pp. 255 LP – 260, Jul. 2015, doi: 10.1126/science.aaa8415.

[18] “AdaBoost Classifier in Python - DataCamp.” https://www.datacamp.com/community/tutorials/adaboost-classifier-python (accessed Jun. 05, 2021).

[19] “How to Develop a Random Subspace Ensemble With Python.” https://machinelearningmastery.com/random-subspace-ensemble-with-python/ (accessed Jun. 05, 2021).

[20] “KNN Classification using Scikit-learn - DataCamp.” https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn?utm_source=adwords_ppc&utm_campaignid=898687156&utm_adgroupid=48947256715&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034352&utm_targetid=aud-299261629574:dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=20880&gclid=Cj0KCQjw--GFBhDeARIsACH_kdauP1-

JkBo1gPf26ducBzYyTBfkdQEP60eWd3hPUno8rJV69UujKuoaArcCEALw_wcB (accessed Jun. 05, 2021).

[21] "UCI Machine Learning Repository: Diabetes Data Set." https://archive.ics.uci.edu/ml/datasets/diabetes (accessed Jun. 05, 2021).

Anexos

Anexo 1: Código do classificador *AdaBoost Tree*.

AdaBoost_class.py | BaggedTree_class.py | SubspaceKNN_class.py

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import pandas as pd
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

#Boosted Tree
from sklearn.ensemble import AdaBoostClassifier

#Ler datasets:
data = pd.read_csv("P68_treino.csv", sep=';')
new =pd.read_csv("P68_teste.csv", sep=';')

#Extrair inputs e target:
X = data.iloc[:,0:12]  #independent columns
y = data.iloc[:,-1]    #target column i.e price range
Xnew_p = new.iloc[:,0:12]
ynew_p = new.iloc[:,-1]

#Construir modelo:
Adaboost = AdaBoostClassifier(n_estimators=80, learning_rate=1, algorithm='SAMME')
modelo = Adaboost.fit(X, y)
prediction=modelo.predict(Xnew_p)

#Avaliação do modelo:
confusion = confusion_matrix(ynew_p, prediction)
print('Confusion Matrix\n')
print(confusion)
print('\nAccuracy: {:.2f}\n'.format(accuracy_score(ynew_p, prediction)))
print(classification_report(ynew_p, prediction))
```