CS434 Final Project Report

Daniel Kato & Nathan Shepherd

2018-06-13

1 Feature formulation and preprocessing

1.1 Features

What are the features you feed to your learning algorithm? Did you simply flatten the 7 rows into a vector for features? Did you transform or aggregate the given data to engineer your own features?

For all of the algorithms used we simply used the flattened 7 rows to predict the event. We used the features provided, and did not use any methods to increase the dimension of the features. For both the K-Nearest Neighbor and the Isolation Forest algorithms, we normalized the data, to reduce noise.

1.2 Preprocessing

Did you pre-process your data in any way? This can be for the purpose of reducing dimension, or reducing noise, or balancing the class distribution. Be clear about what you exactly did. The criterion is to allow others to replicate your works.

For preprocessing our data, we focused on the disparity of normal data to anomalous data. All of the algorithms use subsampling of the test data, to increase the rate of actual events. For the Isolation Forest, we only trained it on the normal non-event data, so it would detect the positive events as an anomaly. For our subsampling, we used this simple line:

```
self.subsample_rate = 0.015 if type == 'general' else 0.1
```

The difference between the general and individual subsampling rates is because of the sheer size of the general data. It should be noted that our subsampling skips over any hypoglycemic events, because these are the rare ones.

2 Learning algorithms

2.1 Algorithms explored

Provide a list of learning algorithms that you explored for this project. For each algorithm, briefly justify your rationale for choosing this algorithm.

- KNN: We used this algorithm on the basis that the hypoglycemic would form a cluster. This was used only on the general data.
- Isolation Forest: This algorithm was chosen because it can be used a small amount of features to find anomalous data. Because there are so many more normal events than hypoglycemic events, we thought that the hypoglycemic events could be found as anomalies, and predicted by how strong of an outlier they were. We used it just for the individual data.
- SVM: Because KNN has the potential to overfit to the training data, we thought an SVM may be able to make a more generalizable model.
- Neural Net: We used a neural network because of it's ability to accurately classify higher dimensional data.

2.2 Final models

What are the final models that produced your submitted test predictions?
We chose to use KNN, SVM, and Neural networks to classify the general data and Isolation Forest, SVM, and Neural networks to classify the individual data.

3 Parameter Tuning and Model Selection

3.1 Parameter Tuning

What parameters did you tune for your models? How do you perform the parameter tuning?

Over the various models used, we tuned different parameters. For each parameter we took a sampling of different values, and selected the one to use based off of performance. This performance was often the F1 value, as that was the final testing value anyway. For the Isolation Forest, we changed the contamination value, which had an effect on how many events it would predict. We raised this until we saw a similar output ratio as the other

predictions. For the Neural Network, we changed the amount of interior nodes. Different values were attempted, but we settled on two layers of 200 nodes each. For KNN, we ran it until an optimal k value was found, based on the improvement rate of testing data. On the Support Vector Machine, we altered the C penalty value, the degree, and kernel type. Using a poly kernel, and second degree with a error penalty of 0.07 worked the best, after testing on various values.

3.2 Model selection

How did you decide which models to use to produce the final predictions? Do you use cross-validation or hold-out for model selection? When you split the data for validation, is it fully random or special consideration went into forming the folds? What criterion is used to select the models?

We used the leave-one-out cross-validation-error to test the accuracy of each KNN model and chose the k with the highest accuracy. With the neural network and support vector machine, we manually tested different combinations of hyper parameters until the accuracy of classifying the test set was highest. We split the data 90%/10% for the training/testing data. These were split the same way each time to allow us to test hyper parameters accurately although, randomizing the split could have provided us with a more robust model.

4 Results

Do you have any internal evaluation results you want to report?
The model with the highest results on the testing data was the neural network with a precision of 0.529, a recall of 0.9, a F1 of 0.666, and an AUC of 0.295. These results were obtained using the neural network with 3 hidden layers with sizes 200, 200, and 10 on the general data.