Context Prediction with Smart Devices

Ohad Edelstein, Nitzan Katz

IDC, Israel

Abstract

Predicting an individual's context (Location, activity, state and so) is beneficial in many fields, from lifestyle monitoring for healthier behaviour, to addiction prevention and rehabilitation, the ability will be a big advancement in tracking applications. This issue was addressed in the ExtraSensory Dataset, where a set of information from smart devices sensors was collected and labeled as a base for farther research in predicting such context.

The Problem in More Details

The issue was tested before[**vaizman2017recognizing**], with standard logistic regression based machine learning technique, the data of 60 participants with over hundreds of thousands of samples were examined by looking at each sample individually, and adjusting the prediction accordingly. This method worked pretty well, and in about 80% of the cases it predicted the activity of the person correctly.
While those results are pretty good, we believe they can be much better. By looking at the sample with relation to the previous samples, we think we can predict better the current state of the person. With a basic logic that, for example, if a person was exercising for the past 10 minutes, he can't possibly be sleeping a minute later. And by looking at the previous samples we can tell the person's context in a more accurate way.

Method

In our work, we followed the same method used in the previous work[**vaizman2017recognizing**], only we split the data into a sets of 10 following samples, and used Recurrent Neural Network so it could be inserted continuously with samples and give an output that considers past samples as well, and not just the current one. For our convenience and time schedules for the work, we focused on only one context prediction out of the 51 tested in the original paper, referenced as "Or_Indoor", which predict if the individual is inside a building or not. After teaching our model using the dataset we created, we test the predictions against the original prediction of the paper regarding this specific context prediction with the expectation for better results.

Results

At first, the original results of the paper were to reproduced. Using the same method, splitting to the same 5-folds groups, were were able to achieve results that seems even better than those of the original paper. We used the method called LFL (Late fusion with learned weights), where the first the prediction is made using one sensor at a time, and then using the results of the prediction from each sensor and combining them in a weighted manner. The way used to measure the success of the model was done with BA scoring, where they took the average of the percentage of correct positive predictions and correct negative predictions. For the data processing, we first filtered empty rows from the dataset, and then made sequences of 10 following samples from each participant. Then the RNN model, we took inspiration from an example available online on https://stackabuse.com/time-series-prediction-using-lstm-with-pytorch-in-python/flights sales prediction (https://stackabuse.com/time-series-prediction-using-lstm-with-pytorch-in-python/). We created an LSTM based model, with a hidden layer of size 100, which we selected arbitrarily to test our model with. We used MSE loss function and Adam optimizer. For the sake of sticking to the example to some level, and with our base knowledge on neural networks, we limited the model to output just one label. We chose the "OR_indoors" label as it was the label with the most samples having this label initiated. We used the same 5-folds division to have 5 different trained models accordingly. It was done so we'd make sure the dataset used to train the model remain the same. A major difference between this model to the original one from the design point of view, is that we didn't split the process into 2 layers of prediction, a prediction per sensor and then a prediction according to the sensor's predictions. We just gave as input all the samples from all the sensors to predict the specific label. Each fold was trained on 50 epochs over the data, then we took the average of the results of the folds for the label. At each epoch we saved the network using pickle, for the network state to remain even if we lose running context in the notebook. The RNN model is better than the original one is some aspects, but worse in other (as seen in figure 1). The BA score which was the main indication for one model being better than another resulted better for the original model. But when looking at Accuracy, Recall and F1 scores the RNN model resulted in better results.

RNN_vs_Reproduction.PNG

Conclusions

In the time given for the work we only managed to test one label, but to prove the improvement our model provides it is essential to test the method on the rest of the labels as well. We might even improve the results farther if we'd follow the LFL design given in the original paper. Although making a more complex neural network should be essentially the same a a bigger network that receives all the sensors inputs at once. So it might not make a significant change on the performance of the model. It's also possible to better collect the data by forcing sample sequences to have a defined maximum time between samples, we just took 10 following samples, but there are some cases in the data where the time passed between two following samples is much greater than a minute (few hours and more!). Also adjusting the length of sample sequences can make predictions better or worse, the length 10 that we chose was convenient for us given the resources we had, but it doesn't have to be, and probably isn't, the best length for sequences for training the network. It's also possible to reconsider the loss function and optimizer used for the network, as we used the ones from the example we went with and didn't experiment with other known available types.

References