EE 373A Project Progress Report 1

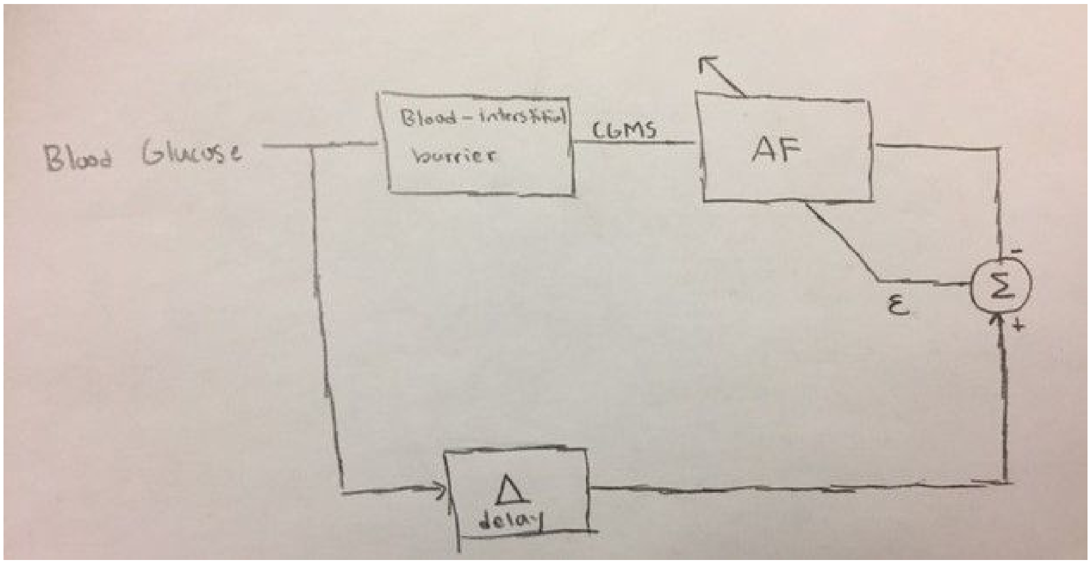
Christian Choe, Min Cheol Kim

Introduction

For our project we drew inspiration from the work of Facchinetti [1]. Continuous glucose monitoring (CGM) device enable diabetic patients to have more control of their glucose level by giving constant readings throughout the day. With constant measurements, patients can easily determine whether their glucose level is too high or too low. However, CGM devices suffer from from poor accuracy and precision even with periodic calibration with a blood glucose measurement (BGM). Since CGM devices measure subcutaneous glucose levels and not blood glucose level, the readings should be processed in order to achieve more accurate readings. In order to improve the reliability of CGM devices we are using an adaptive signal processing approach by testing various methods discussed in class.

Model

The block diagram below depicts the signal processing scheme of the problem. The input signal is blood glucose which travels through the blood-interstitial barrier giving rise to the continuous glucose measurements. We are trying to design an adaptive filter to transform the continuous glucose measurements to blood glucose measurements. The adaptive filter will be the inverse of the blood-interstitial barrier filter. Since the continuous glucose measurement typically lags behind the blood glucose measurements, we will incorporate a delay when comparing the output with our desired signal. From inspection of our data, the delay seems to vary from 10 to 30 minutes.



Data

The data we are using is from a clinical trial studying the accuracy of the glucose watch monitoring device compared to Medtronic continuous glucose monitoring device. We have data for 103 patients. For each patient we have time points for blood glucose measurements, continuous glucose measurements, and glucose watch measurements. Since we are focusing on improving the reliability of continuous glucose measurement we are ignoring the glucose watch data. The continuous glucose measurements are taken every 5 minutes and the blood glucose measurements are taken every 30 minutes. In order to facilitate the training of our adaptive filter we interpolated the blood glucose measurement using linear interpolation followed by a gaussian blur. We used the interpolated blood glucose measurement as our desired signal (gold standard).

Every patient in the study had either one or two CGM devices attached either on the abdomen or the buttocks. Since we could not discern a noticeable difference between CGM readings with respect to CGM device location, we use the data indiscriminately. If a patient has two CGM devices, we simply take the CGM data from the device that has more data points.

Experiments & Results

For each technique, we conducted the following experiments:

Single patient experiment. In this approach, we treated each patient as both training and testing signal. We initially simply trained the filter the whole time and plotted the current filtered signal. In a second attempt, we halted training at half point of the patient’s data and tried to see if the filter generalized to the second half.

Multiple patient modeling experiment. In this approach, we trained a filter across signals from multiple patients to train a filter that would, ideally, generalized to a new patient. We divided up our data into training and testing patients, and the adaptive filter was trained on all the data from the training patients. Afterwards, the trained filters were tested on the unseen, new patients in the testing patient set.

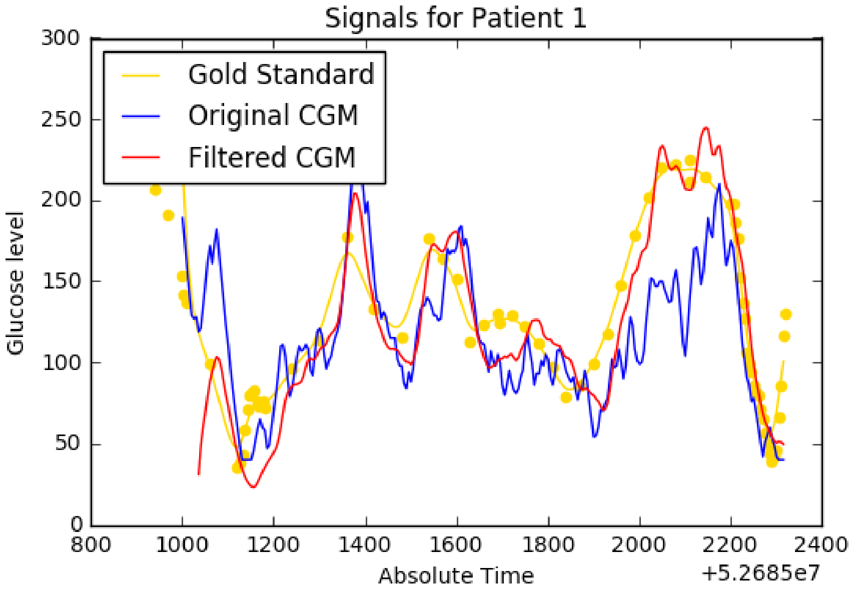
Optimizing Filter Hyperparameters

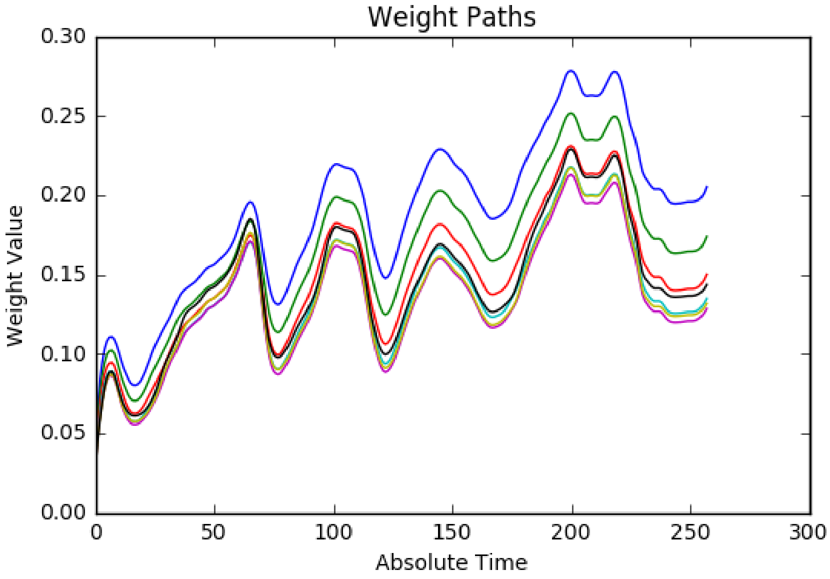
The metric for measuring the similarity in two glucose measurement signals in literature was the mean absolute relative difference (MARD) [1]. For the multiple patient modeling experiment, we measured the MARD between the filtered signal and the gold standard, as well as the MARD between the original CGM signal and the gold standard. Smaller the MARD, the closer the signal is to the gold standard. Using MARD as a guide, we sweeped a range of hyperparameters for our adaptive filter to find the optimal one.

mu-LMS Filter

Single patient experiment: We see that in general, the filter is able to fit the gold standard very well while training. This is most likely due to the fact that the filter can pick up local variations in the original CGM signal. When the adaptation is stopped at half, however, the filter does not seem to generalize well.

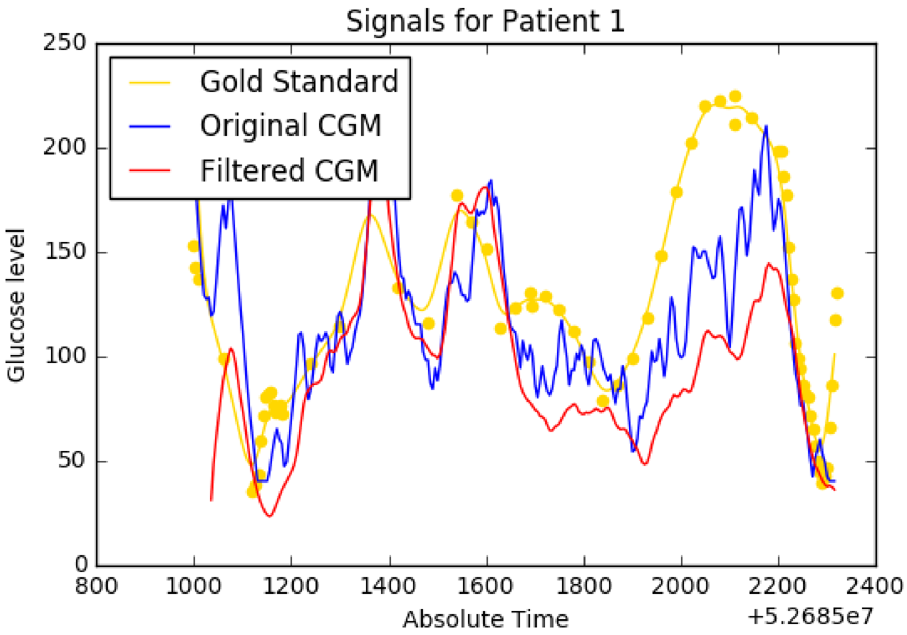
The following is an example plot of the filtered signal (red), desired signal (gold), and the original input signal (blue), as well as the trajectory of the weights. From the trajectory of the weights, we can see that convergence does not really occur, and the weights seem to be adapting to the local conditions of the signal.

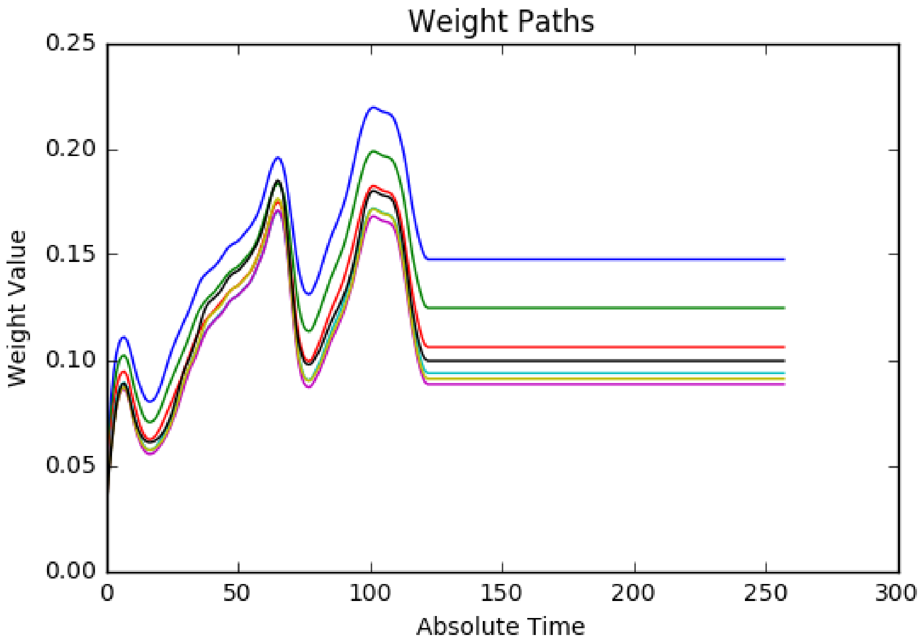




The following plots show the same signals and the weight values, except that training was halted at the halfway point of the signal. We can see that the filter does not generalize in the second half, which makes sense since the original weights did not converge by the halfway point. One way to remedy this issue is to explore more possible mu values and see if convergence occurs earlier, but it does not seem like any mu value would actually reach convergence in this case.

We discuss data from Patient 1 as an example, but all of the discussions above were generally true for all patients we looked at.





Multiple patient modeling experiment: We used 70% of the patients we have the data for as the training set for this type of experiment (for every technique). We trained mu LMS filter on the original signals and gold standard signals from every patient in the training set, and see if the filter produced signals that are better than original in the testing set.

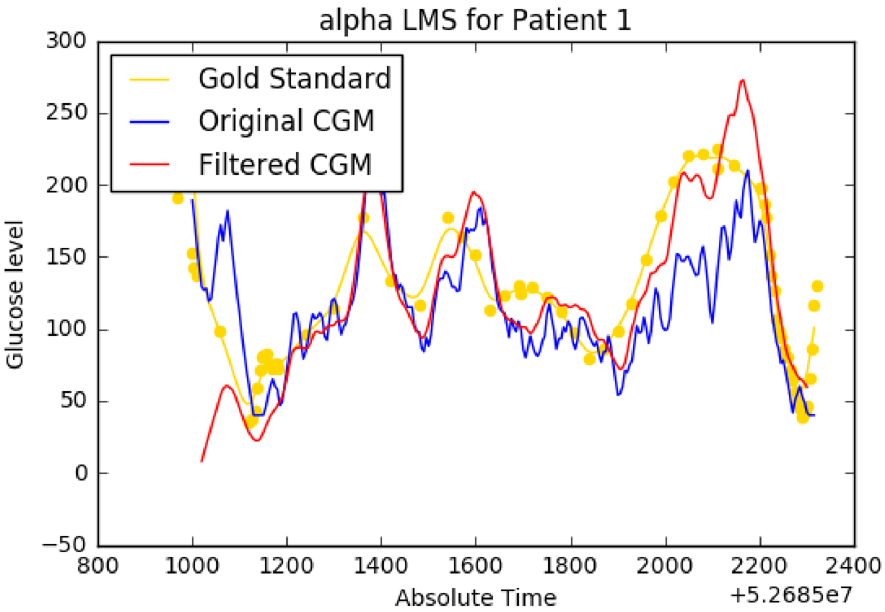
We found that the overall MARD between the original CGM signal and the gold standard is 0.234.

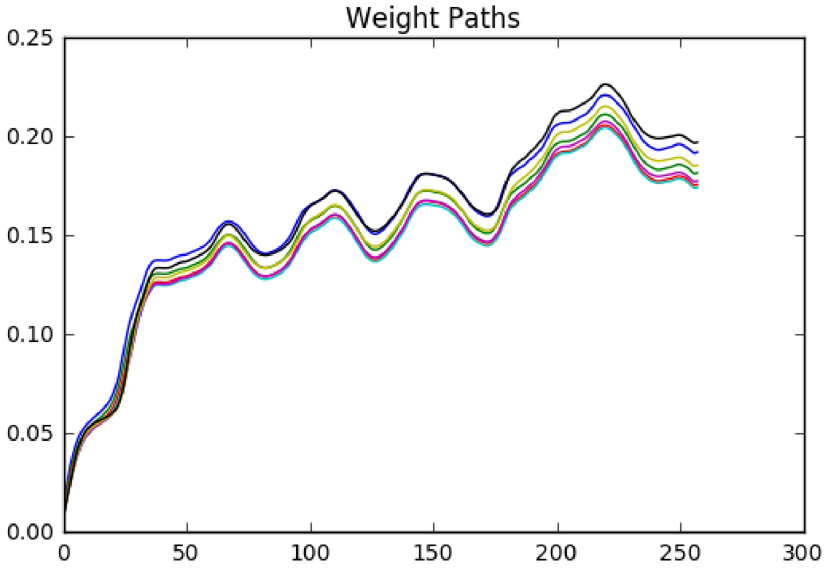
For mu-LMS, despite experimenting with different hyperparameters (fraction of mu_max, number of times to repeat through the training data, etc), we got a MARD for the filtered signal that was only marginally better than the MARD for the original CGM signal. The best overall MARD between the filtered signal and the gold standard in the testing set was 0.223, using 5 weights and mu = 0.1*mu_max. This corresponds to about a 1% improvement in MARD, which is a percentage metric.

.

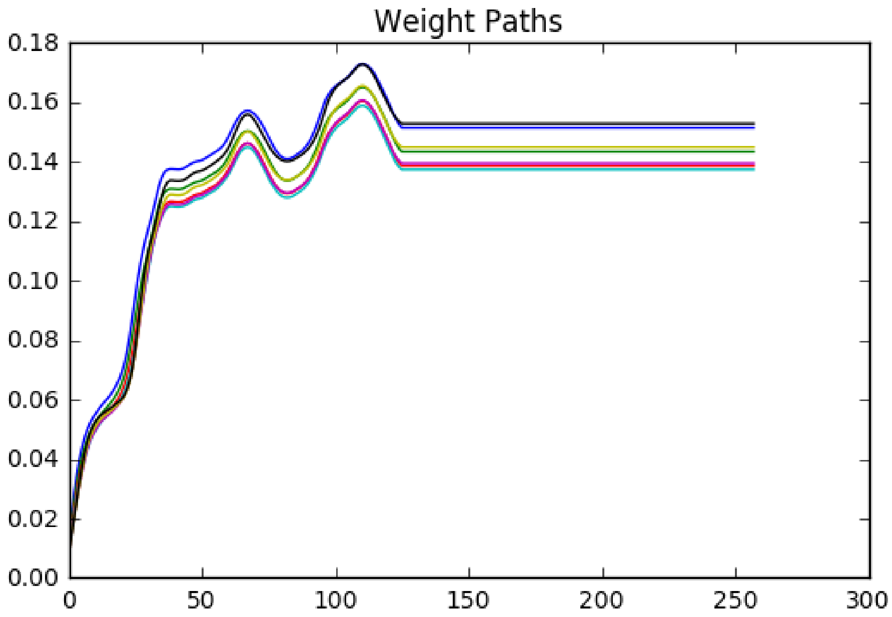
alpha-LMS Filter

Single patient experiment: Similar to mu-LMS filter, we saw that the filtered signal closely follows the gold standard signal during training. We noted that the alpha-LMS filter is a lower variance filter, as it does not vary wildly as the mu LMS technique; even though the weights do not seem to reach convergence, it seems to overfit the training signal less. We have similar plots as above for patient one, for the case where the filter was training the entire time.





When the training for the filter is stopped mid-way, the fit of the filtered signal to the gold standard became much worse. Plots for this experiment follows.



Multiple patient modeling experiment: We used the same training set as in mu LMS. We trained alpha LMS filter on the original signals and gold standard signals from every patient in the training set, and see if the filter produced signals that are better than original in the testing set.

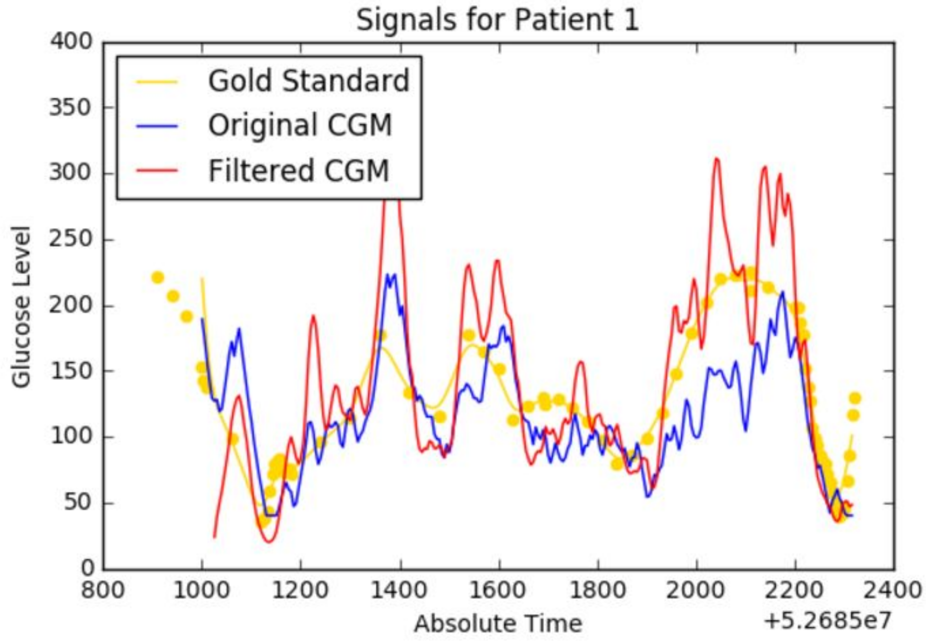
For alpha-LMS, we experimented with different hyperparameters (fraction of mu_max, number of times to repeat through the training data, etc), and in general we got a MARD for the filtered signal that was similar to that of the original CGM signal. The best overall MARD between the filtered signal and the gold standard in the testing set was 0.22. This corresponds to about a 1.4% improvement in MARD.

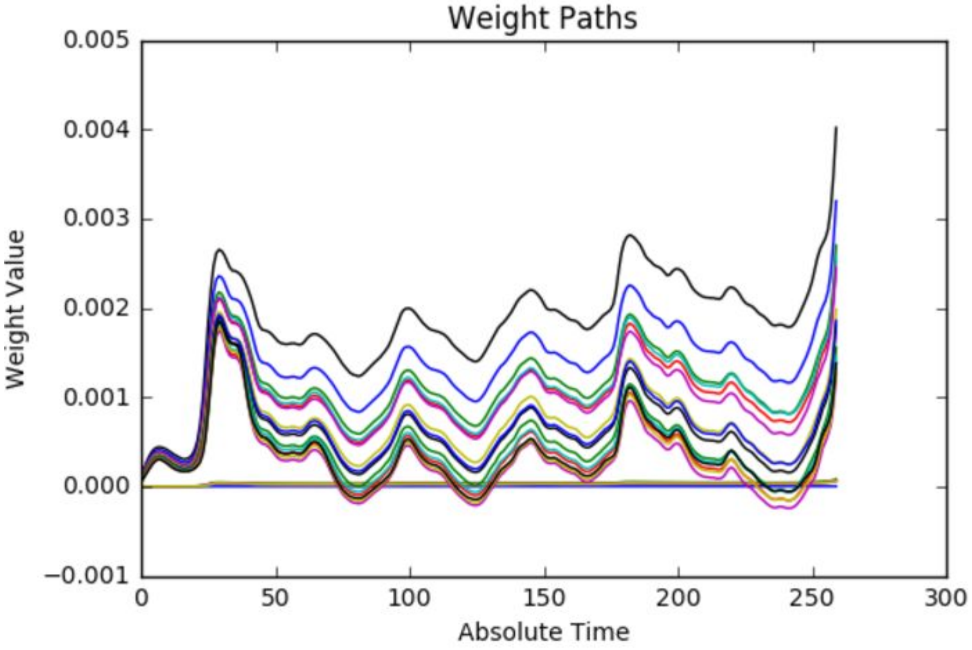
Volterra alpha-LMS Filter

Notes on implementation: We used a truncated Volterra series with L = 2. (We also tested L > 2, but the results had too much variance to be meaningful)

Single patient experiment: In almost all the patients, we see that while the filter is able to follow the trends of the gold standard it is much noisier than mu or alpha LMS. This is probably because the cross terms and squared terms increases the power of the noise as well. When examining how the weights adapt, its interesting to note that some weights are almost zero.

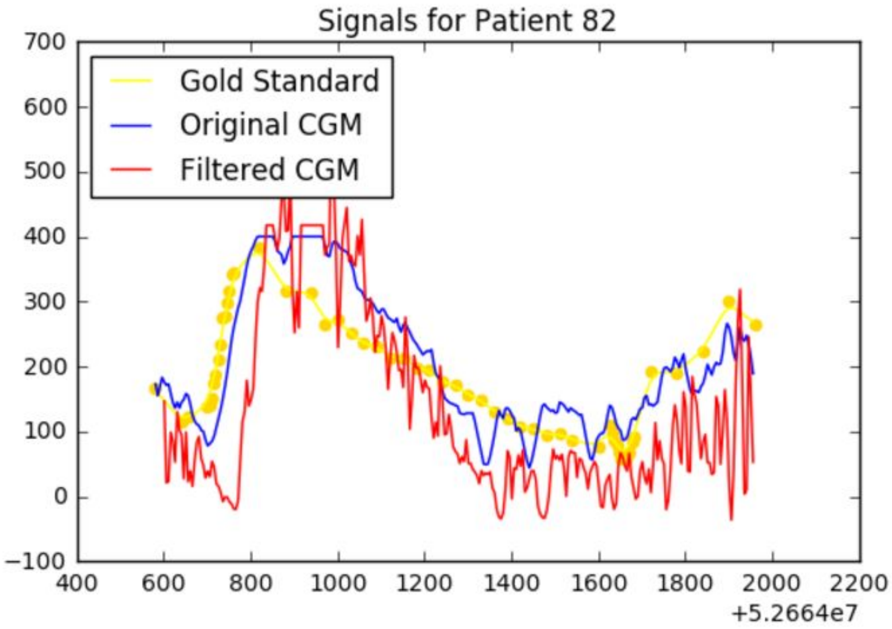
The following is an example plot of the filtered signal (red), desired signal (gold), and the original input signal (blue), as well as the trajectory of the weights. From the trajectory of the weights, we can see that convergence does not really occur as seen in using LMS, and the weights seem to be adapting to the local conditions of the signal. Compared to LMS, the filtered signal has much higher variance.





Mutliple patient modeling experiment: We used 70% of the patients we have the data for as the training set for this type of experiment (for every technique). We trained Volterra alpha-LMS filter on the original signals and gold standard signals from every patient in the training set, and see if the filter produced signals that are better than original in the testing set. After training we made the weights in the adaptive filter to be constants unlike the single patient experiment.

As seen in the plot below the filtered CGM does not seem to do any better than the original CGM data. In fact, due to the high variance, the filtered CGM does worse.



For Volterra alpha-LMS, we experimented with different hyperparameters (fraction of alpha_max, number of times to repeat through the training data, etc). The best overall MARD between the filtered signal and the gold standard in the testing set was 1.153. This corresponds to about a 91.79% worsening in MARD.

Note on Causal vs Non-Causal Filters

We tried everything we presented above with noncausal filters as well, the filters whose weights get information from future data points. This did not seem to help much, even with different filter lengths and delay models.

Next Step

We plan to implement a neural network. We can optimize the parameters for training the adaptive filter by using the MARD score but it may not help since the weights do not converge.

Github

All of our code is available on Github. We used python for this project.

https://github.com/mincheoly/adaptive_cgm

Citation

[1] Facchinetti A, Sparacino G, Cobelli C. Signal processing algorithms implementing the "smart sensor" concept to improve continuous glucose monitoring in diabetes. J Diabetes Sci Technol. 2013;7(5):1308-18.