IFT 6269 project

Drew Kristensen, Antoine Chehire, Adel Nabli

November 2018

1. Introduction:

Electroencephalography (EEG) is a widely used method to capture the electrical activity of the human brain as it is simple and non-invasive (we only need to put some electrodes over a subject's head). The drawback being that the data captured is often a mix of several signals: the informative cortically generated signal (which we want to capture) is contaminated by other cerebral electrical artifacts such as electromyography (EMG) or electrooculography (EOG). In addition, neuroscientists can expect an additive noise coming from the electronics used to capture the signals. Thus, in order to use the EEG data, the first step is often to clean it. As EEG data comes naturally in the form of a time series, one way of looking at this task is saying that we want to perform inference to find the filtering distribution over some directed graphical model where the observed variables are noisy.

We will try to make several sets of assumptions over the properties of our graphical model and try different methods. Then, having built those methods, we'll have to assess them.
The problem is that to compare the results of a denoising methods to the ground truth, we need to have access to both a plausible ground truth and a plausible noisy data. But datasets containing both noisy and clean data (i.e only containing the informative cortically generated signal) don't seem to be easily found. Most of the papers assessing denoising methods on this kind of data either have access to complying patients that are ordered for a moment to be immobile during the recording, or model the two kinds of EEG using real signal characteristics. But that last approach seems to require a lot of field knowledge that we don't have.

Thus, what we decided to do instead is to assess our methods as a *pre-processing* step in a bigger problem, which is often the case in reality (for example in the *Brain-Computer Interface* field, we might want to do a classification task over the EEG data). Hence, the *"best"* method would be the one leading to the best results in the bigger task.

We applied our methods to the openly available **Grasp-and-Lift EEG Detection** dataset which contains the recordings of 12 subjects as they perform 6 types of simple tasks (ex: grasp and lift an object with one hand).

2. Related Works

This is not the first approach to this problem, as EEG data is inherently noisy. Furthermore, there is no "true data" available to us, which presents yet another problem. In Geraldo et. al [1], they used a Kalman filter to recreate the spatial activity within the brain from EEG data using physiological models with a normal Kalman filter and a non-linear extension. This proved that for this application, Kalman filters performed well. However, they did no classification on the data, their paper only sought to recreate the temporal-spatial activity observed from the data. A similar task was accomplished in Lamus et al [2] using more sophisticated methods in conjunction with a Kalman filter for recreated the spatial activation from EEG data, both simulated and human-generated. While the task relied on other techniques to help the Kalman filter, they showed promising results from their model. Again, this work did not attempt any sort of classification or comparison of the data generated by their model compared to the raw data. In order to get around the difficulty of lacking our ground truth for the data, multiple studies have used simulated EEG data by mixing many samples of 2kHz data together to create an EEG like signal. In Salis et al [3], they compared the effects of a Kalman filter to both Empirical Mode Decomposition and Discrete Wavelet Transforms in estimating the true states of this noisy simulated data. They found that Wavelet Transforms performed the best, with Kalman filters being the second best option. However, they note that all three of their methods resulted in a significant decrease in artifacts from their data. In Vorobyov and Cichocki [4], they demonstrate the ability of ICA in extracting the true signals from this type of simulated data. They used ICA with subspace filtering to clean the data before projecting their results back onto the sensor-level. This gave them clearer representations of the data than before they started. Again, this study did not attempt classification with this data, as it was simulated.

3. Models

3.1. Kalman Filter

A Kalman filter is an extension of a Markov Chain which allows for a gaussian noise component, modeling the noise observed in real applications. This technique proves to be better than using raw data as the Markov Chain component means that we are modeling the joint probabilities at each time step. The Kalman filter works in an iterative process, first it estimates the new state, then it updates the values based on what the actual observed values were. Let $\mathbf{A}_k$ be the transition model from state $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ for arbitrary k. Let $\mathbf{H}_k$ be the observation model. Let $\mathbf{Q}_k$ be the covariance for the process noise, collected by the sensor, with $\mathbf{R}_k$ be the covariance of the observation noise. Let $\mathbf{w}_k$ be the noise vector drawn from $N(0, \mathbf{Q}_k)$. For our application, we disregard the control-input components. Then, we have that given our previous state $\mathbf{x}_{k-1}$,

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k$$

This gives us the observation at time k. To find the state, $\mathbf{z}_k$ given the previous time step, we have

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{v}_k$ is the noise of our process, which is a vector given by $N(0, \mathbf{R}_k)$

We can use this filter to obtain good estimates of the underlying hidden state, which had been obscured by noise in both the observation and process. Examples of this better estimate can be seen in figure 3.

3.2. Independent Component Analysis

Independent Component Analysis (ICA) is a technique to separate independent sources of data in a multivariate signal. This has been used extensively in signal processing, as it is a good baseline method to separate out signals which may be interfering with each other. In our case, we assume that the multivariate signal data we are receiving is a combination of the activations across all the different regions of the brain which we are monitoring. To use this algorithm, we have to assume that our data fits two criterion. First, our signals should not be Gaussian in distribution. If they were, a simple Gaussian Mixture Model may handle this example better. Secondly, we assume that our signals are independent of each other within the multivariate signal we observe. With EEG data, this may not be the case, as it is difficult to know which activations within the brain are linked to each other, and which ones are not. However, we relax this assumption in order to test on our data. Our goal with ICA is to separate our signals in such a way that we minimize the amount of information shared between any two components and also maximize the difference between our signal and a Gaussian signal. To accomplish this, we assume that our observed signal $\mathbf{x}$ can be written as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, with each x_i being the value we observed at time i. As we assume this is a mixture of signals, we can then define $\mathbf{s}$ as our vector of source signals, with $\mathbf{s} = (s_1, s_2, \ldots, s_k)$. These source signals are weighted with values from $\mathbf{a}_i$, $\mathbf{a}_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,k})$. This allows us to express each x_i as

$$x_i = \sum_{j=1}^{k} s_k a_{i,k} = \langle \mathbf{s}, \mathbf{a}_i \rangle.$$

We can extend this definition to model the full multivariate signal we observe. If we define a matrix $\mathbf{A}$ to be $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$, then

$$\mathbf{x} = \mathbf{A}\mathbf{s}.$$

From this, if we multiply both sides by the inverse of $\mathbf{A}$, or psuedoinverse if $\mathbf{A}$ is not square, then we can recover the source signals from the observed data with

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}.$$

Furthermore, we can add in a noise component, $\mathbf{n}$, which we assume to be Gaussian with mean 0 and with a diagonal covariance matrix such that $\mathbf{s} = \mathbf{A}^{-1}(\mathbf{x} - \mathbf{n})$. We will use ICA in order to compare the difference a Kalman filter makes compared to using the unfiltered data.

4. Experiments

4.1. Method

As explained in the introduction, it is very difficult to asses the effectiveness of a filtering algorithm directly. Therefore, we decided to focus on the effect of our filtering on a downstream task.

For the downstream task, we chose to study classification as it is a common practice. More precisely, we first filter the data before using this filtered data as input for a classification task.

4.2. Data

For the evaluation of our filtering algorithm, we focused on the **Grasp-and-Lift EEG Detection** dataset. The patients were monitored while performing 6 tasks, mostly about grasping and lifting an object. The recordings were then split in 6 supervised binary classification datasets, one for each task. However, the data couldn't be used directly as over 97 percent of the data is made of negative examples. Thus, most classifiers predicted the class 0 regardless of the input data. This made the comparison of the filtering algorithms impossible as the downstream classifier couldn't measure the impact of the filtering algorithm. Thus our first task was to balance the data.

In order to reach a balance, We chose to subsample only a few negative examples before applying our classifier. We implemented two ways of sub-sampling:

- At random
- Selecting only the samples preceding and following the positive examples

The first strategy gives an approximate idea of the real accuracy of the prediction for a given sample. This is due to the randomness.

The second strategy is, however, much more interesting as it focuses on the transitions. This task is much harder as it is especially difficult to accurately determine the clear separation between two actions. In fact, the raw data shows that the distinction itself isn't that clear as some tasks overlap. For instance, the scientists who recorded the EEG measured data that corresponded to both grasping and touching. Thus, even for the scientists, the distinction isn't that clear.

4.3. Classifier

Our original aim is to measure the impact of filtering algorithms in extracting relevant information from EEG data. Therefore, the choice of the classifier is not that important as what matters is the relative improvement or lack of. In the end, we settled for a Logistic Regression as it was fast and served our purpose as well as any other classifier we tried.

As we were curious, we also tried other classifiers, such as Random Forests or Naive Bayes classifier. They yielded similar accuracy scores in general and didn't change our overall conclusions on the impact of filtering.

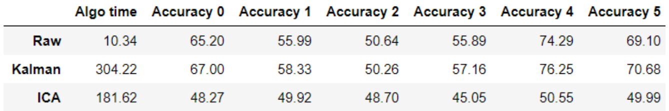
4.4. Results

Following the method described above, we compared the accuracy scores for each of the seven tasks. We evaluated three models:

- Raw: The raw data as is. It serves as a baseline.
- Kalman: Using Kalman filters to generate filtered data.
- ICA: Using ICA to generate filtered data.

We displayed the results using the two strategies for sub-sampling. We used the Logistic Regression as classifier (LogReg) and measured not only the accuracy for each of the 6 tasks, but also the time the algorithm took to filter the data (in seconds).

Using random sub-sampling, we obtain the following results:



	Algo time	Accuracy 0	Accuracy 1	Accuracy 2	Accuracy 3	Accuracy 4	Accuracy 5
Raw	10.34	65.20	55.99	50.64	55.89	74.29	69.10
Kalman	304.22	67.00	58.33	50.26	57.16	76.25	70.68
ICA	181.62	48.27	49.92	48.70	45.05	50.55	49.99

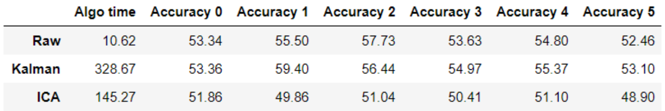
Figure 1: Random sub-sampling

First of all, we can notice the algorithm time for raw data to be non nil. This occurs as we need to load the data in memory. Thus, to get the actual time, we have to subtract approximately 10 seconds.

We also observe that not all tasks are as difficult as Accuracy 2 is around 50 percent while Accuracy 4 is around 75 percent with raw data.

We can see that the ICA model yields the poorest results. This surprised us a lot as it is a technique widely used in competitions. We can also notice that the Kalman filter improves the accuracy scores in every task, but not by a big margin.

We also tried to focus on transitions. We then obtained the following results:



	Algo time	Accuracy 0	Accuracy 1	Accuracy 2	Accuracy 3	Accuracy 4	Accuracy 5
Raw	10.62	53.34	55.50	57.73	53.63	54.80	52.46
Kalman	328.67	53.36	59.40	56.44	54.97	55.37	53.10
ICA	145.27	51.86	49.86	51.04	50.41	51.10	48.90

Figure 2: Transition sub-sampling

As we can see, the task is much more difficult as the accuracy scores dropped by a lot when using the raw data or applying a Kalman filter. We still notice a slight improvement with Kalman filters compared to using raw data, but nothing too impressive.

Oddly enough, the ICA accuracy scores improved on average. It seems the algorithm is better when dealing with transition. We believe this to be the most likely explanation as to why the ICA is nearly always used in the best solutions in competitions. However, it still remains the worst model when used on its own.

5. Conclusion

This work presented the impact of using graphical models in filtering EEG data. As presented in the results section, some models may have some positive impacts when defining transitions while others might be better overall. It also seem to depend on the task as some tasks are easier than others.

This work evaluated the algorithm on only one dataset. It is thus difficult to draw conclusions on how efficient filtering is in general. On top of that, the labels may not be exact as explained in section 4.2 as we noticed overlaps in labels. We tried to find other datasets to give a more general perspective to our study, but finding labelled data proved very had. We had lots of unlabelled data though, but we couldn't use it to evaluate the quality of our filtering.

As we were curious, we looked at the best models used in competitions. It turns out to be a fine tuning of a mix of ICA and Kalman filters. Unfortunately, we didn't have time to delve more deeply in the fine tuning.

References

[1] E. Giraldo, A. J. den Dekker, and G. Castellanos-Dominguez. "Estimation of

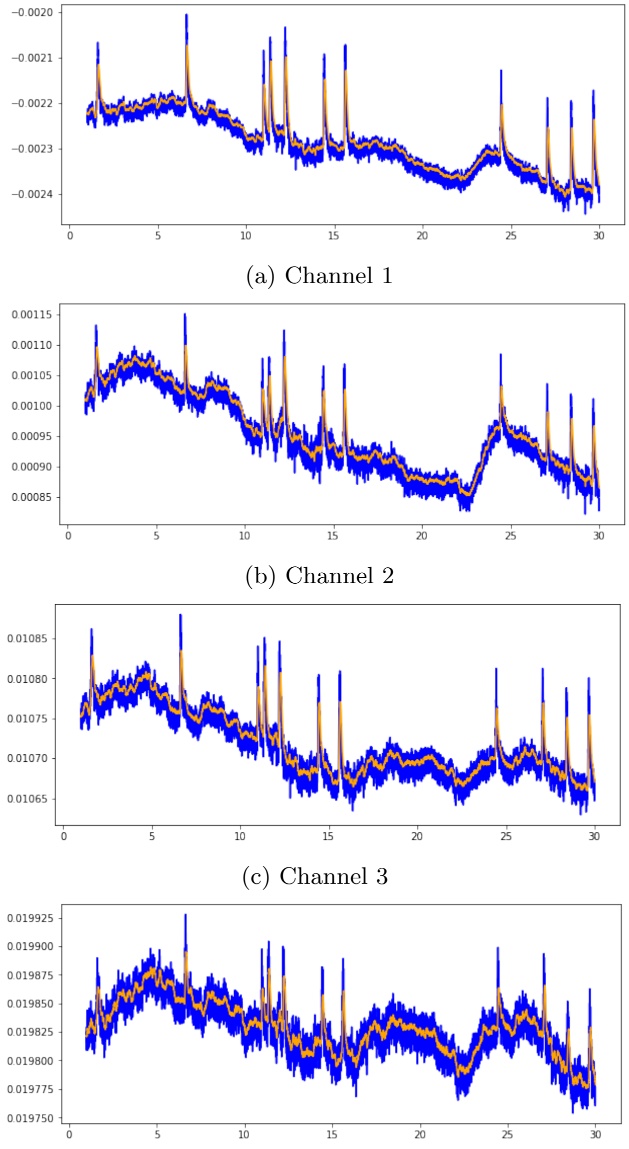
dynamic neural activity using a Kalman filter approach based on physiological models". In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. Aug. 2010, pp. 2914–2917. DOI: `10.1109/IEMBS.2010.5626281`.

[2] Camilo Lamus et al. "A Spatiotemporal Dynamic Solution to the MEG Inverse Problem: An Empirical Bayes Approach". In: *arXiv e-prints*, arXiv:1511.05056 (Nov. 2015), arXiv:1511.05056. arXiv: `1511.05056 [stat.AP]`.

[3] C. I. Salis et al. "Denoising simulated EEG signals: A comparative study of EMD, wavelet transform and Kalman filter". In: *13th IEEE International Conference on BioInformatics and BioEngineering*. Nov. 2013, pp. 1–4. DOI: `10.1109/BIBE.2013.6701613`.

[4] Sergiy Vorobyov and Andrzej Cichocki. "Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis". In: *Biological cybernetics* 86 (May 2002), pp. 293–303. DOI: `10.1007/s00422-001-0298-6`.

6. Appendix



(a) Channel 1

(b) Channel 2

(c) Channel 3

(d) Channel 4

Figure 3: The blue lines represent the raw EEG data for the first subject, and the orange is the predicted path from our Kalman Filter. Each figure shows 30 seconds of data from the OpenMIRR dataset.