Empirical Priors Report

Jason Ma

January 2019

1 Step 1

Changes are reflected in the Github repository.

2 Step 2

In this step, we explore the end-of-day(EOD) self-report's potential bias and variance as an estimator for the smoking times reported in event-contingent EMAs. That is, treating event-contingent EMAs as ground truth, how accurate are the EOD EMAs?

eod_bias_variance.ipynb contains our code. Essentially, we compute bias and variance based on a window of time within which a signal(an hour window in which the participant indicates that he smoked) in the EOD EMA is treated as an estimator for the corresponding smoking time reported in the event-contingent EMA. For example, suppose a user reported smoking at 3:30pm in the event-contingent EMA and the window parameter is set to 2 hours. Then, only signals between 1:30pm to 5:30pm in the the EOD EMAs are used in computing the bias for this particular smoking time. Smoking times in the event-contingent EMA that do not have any signals in EOD EMAs within the fixed window are treated as "missed", indicating that they are not covered by signals in the EOD EMA.

We note that the signals in the EOD EMA is a range of time as opposed to a timestamp. Then the question is how do we compute the bias? Our appraoch is to convert the hour window into a single timestamp that is simply 30 minutes past the beginning of the window and then perform simple arithmetic to compute the bias. For example, suppose we have a smoking time 2:45pm and our signal is 2:00pm-3:00pm. Then, we convert our signal to 2:30pm, and hence the bias is 15 minutes.

The table below summarizes the bias and variance statistics by setting the window to 1,2,3,4 hours, respectively.

Window(hr)	Mean of bias	Variance	Covered	Missed
1	33.9	649.5	107	72
2	45.3	1424.3	124	55
3	50.6	2050.4	129	50
4	50.6	2050.4	129	50

Table 1: EOD EMA bias variance table

The table above makes logical sense; as the window of time increases, the bias and variance increase because signals farther away in time are also considered. Consequently, the number of smoking times covered also increases. We see that the increase in mean bias is much greater from 1 to 2 than from 2 to 3, suggesting that setting the window to 2 hours is probably the most reasonable as it accounts for people's lossy memory while avoiding double-counting/overfitting. If the window parameter is too high, then it is likely that we are double counting signals by matching them with two smoking times in the opposite sides of the signal or two smoking times that are super close to one another. Therefore, I believe that 2 hour window is appropriate, and in this case, the bias is **45** minutes.

3 Step 3

4 Step 4

In this step, we investigate the temporal alignment of days of the participants. It is understood that the hours of use don't match across individuals, but it is still important to understand the degree to which they differ. In particular, we are interested in the following two questions:

- Do most people when they use the system, last the 12 hours that the system is supposed to be on?
- Do people's hours of use change wildly over the course of the study?

alignment.py, day_alignment.py, and day_alignment.ipynb contain our code for this step. Essentially, we first collect and clean the start and the end of system timestamps for each participant from their respective start_day and end_day csv files. Then, we compute the temporal alignment for each day for each user and record them in alignment_user.csv. Note that there are a few users(202,204,210,212) whose start_day or end_day file is empty, and therefore they are not included in the analysis. Then, we compute the mean, standard deviation, max, min of temporal alignments for each user and record them in alignment_summary.csv. Finally, we compute the overall mean and variance of temporal alignments across the users.

This table, which is screenshotted from alignment_summary.csv, contains the summary statistics of the temporal alignments across the users.

participant_id	mean	stdev	max	min	entries
201	10.7372296	2.731537362	12.80171472	7.639866667	3
203	10.8133214	2.092461943	13.13506306	8.068243889	4
205	9.73696707	1.219089037	13.10959444	8.294276389	13
206	8.72122978	0.851911811	9.819039722	8.00512	5
207	13.9568006	0.290896961	14.24670417	13.4703875	6
208	10.2601099	1.833793879	12.06064667	7.707658333	4
209	9.37013694	1.114114386	10.09445333	8.087231667	3
211	10.3557156	1.942385919	13.34581083	7.590943056	14
213	12.1021845	3.503399332	14.50571583	8.082425833	3
214	12.2948646	1.705032252	14.76357639	9.550181944	11
215	12.6721328	1.060546962	13.96728556	10.59635806	13
216	11.3815492	2.303332228	14.56213139	8.212066111	13
217	9.66196616	1.926325661	11.59668222	5.525652778	12
218	12.1777603	1.555017615	13.99422861	7.816930556	14
219	11.1593159	1.309386897	12.80970167	8.517146111	14
220	12.5765062	0.864961888	13.18812667	11.96488583	2
221	8.96314007	3.186235066	12.88133722	5.6369875	4
222	10.5476607	2.836644295	12.55347111	8.541850278	2
223	12.3165996	1.234628747	14.01247028	8.771981667	14
224	12.2337675	2.105032178	14.82978917	8.163670278	10
225	12.1304208	1.846080827	14.58822222	8.490066389	10
226	13.6024329	1.098169923	14.91197028	11.606815	13
227	11.8183273	3.579998052	14.45956556	0.314554167	13
228	12.7712154	2.461162316	13.98287389	8.379602778	5
229	13.5394527	1.652764715	14.91433972	11.70575639	3
230	11.3561625	1.422800259	13.91825306	8.587545833	13
231	14.5228792		14.52287917	14.52287917	1
232	7.34267889		7.342678889	7.342678889	1
233	10.5159818	3.430887292	14.1195925	2.217635833	10
234	12.0026526	1.506598273	13.60962639	10.62204806	3
235	12.1173652	0.960730118	13.79403778	11.23666833	7
236	11.913991	3.184147208	14.16552306	9.662458889	2
237	13.8101861	0.360822448	14.06532611	13.55504611	2

Figure 1: Temporal alignment statistics

The overall mean of temporal alignment, which is computed by taking the average of the mean alignment of each user, is **11.5** hours, and the variance is **2.59** hours squared.