Sleep Tracking

Objective

We have to predict the sleep times of users using raw accelerometer data considering sleep as android as the ground truth

Steps Involved

1.Data Collection

2.Data Preprocessing(Converting the format obtained data)

3.Training(Obtaining patterns from Sleep as android data)

4.Testing(Assign states depending on the patterns obtained)

How Sleep as android works?

1.They predict sleep based on accelerometer.They have other options too(sonar)

2.They store 1 value for every 5 min

3.The values they store range from 1 to 10 but in general raw data ranges from approximately 9.2 to 15 or 16 depending on movement of the device

4.They also store something like an event called DHA(data high activity) which represents 0 or 1 for every 10 seconds depending on activity

What we don’t know :(

1.we don’t have any idea at what frequency they collect the data which won’t be a good problem

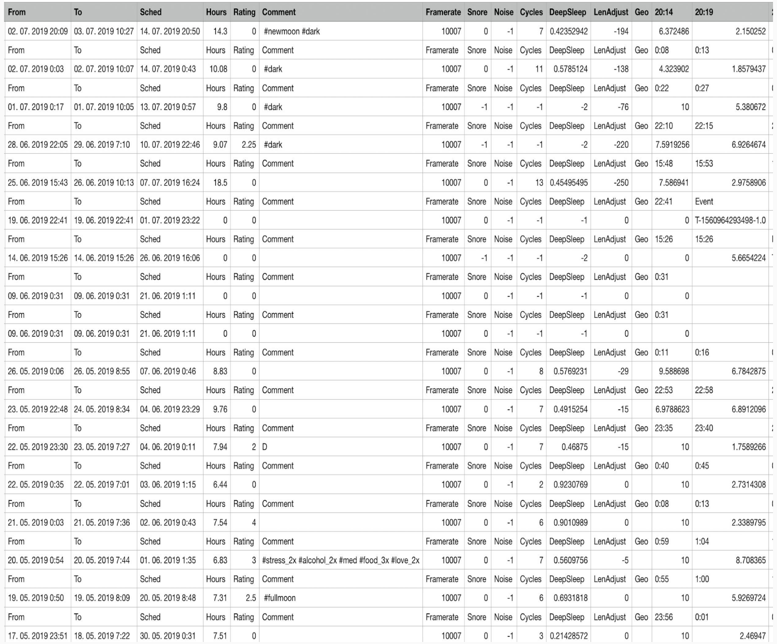
2.Why the values they store range from 1 to 10 which has been the main problem

3.How do they start the service if app isn't started manually.

Data Preprocessing

We extract the start and end times and assign the size of the array depending on no. of windows

We collected data for every 6 sec and aggregated to 5 min by taking the average



From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	20:14	20:19
02. 07. 2019 20:09	03. 07. 2019 10:27	14. 07. 2019 20:50	14.3	0	#newmoon #dark	10007	0	-1	7	0.42352942	-194		6.372486	2.150252
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:08	0:13
02. 07. 2019 0:03	02. 07. 2019 10:07	14. 07. 2019 0:43	10.08	0	#dark	10007	0	-1	11	0.5785124	-138		4.323902	1.8579437
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:22	0:27
01. 07. 2019 0:17	01. 07. 2019 10:05	13. 07. 2019 0:57	9.8	0	#dark	10007	-1	-1	-1	-2	-76		10	5.380672
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	22:10	22:15
28. 06. 2019 22:05	29. 06. 2019 7:10	10. 07. 2019 22:46	9.07	2.25	#dark	10007	-1	-1	-1	-2	-220		7.5919256	6.9264674
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	15:48	15:53
25. 06. 2019 15:43	26. 06. 2019 10:13	07. 07. 2019 16:24	18.5	0		10007	0	-1	13	0.45495495	-250		7.586941	2.9758906
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	22:41	Event
19. 06. 2019 22:41	19. 06. 2019 22:41	01. 07. 2019 23:22	0	0		10007	0	-1	-1	-1	0		0	T-1560964293498-1.0
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	15:26	15:26
14. 06. 2019 15:26	14. 06. 2019 15:26	26. 06. 2019 16:06	0	0		10007	-1	-1	-1	-2	0		0	5.6654224
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:31	
09. 06. 2019 0:31	09. 06. 2019 0:31	21. 06. 2019 1:11	0	0		10007	0	-1	-1	-1	0		0	
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:31	
09. 06. 2019 0:31	09. 06. 2019 0:31	21. 06. 2019 1:11	0	0		10007	0	-1	-1	-1	0		0	
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:11	0:16
26. 05. 2019 0:06	26. 05. 2019 8:55	07. 06. 2019 0:46	8.83	0		10007	0	-1	8	0.5769231	-29		9.588698	6.7842875
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	22:53	22:58
23. 05. 2019 22:48	24. 05. 2019 8:34	04. 06. 2019 23:29	9.76	0		10007	0	-1	7	0.4915254	-15		6.9788623	6.8912096
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	23:35	23:40
22. 05. 2019 23:30	23. 05. 2019 7:27	04. 06. 2019 0:11	7.94	2	D	10007	0	-1	7	0.46875	-15		10	1.7589266
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:40	0:45
22. 05. 2019 0:35	22. 05. 2019 7:01	03. 06. 2019 1:15	6.44	0		10007	0	-1	2	0.9230769	0		10	2.7314308
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:08	0:13
21. 05. 2019 0:03	21. 05. 2019 7:36	02. 06. 2019 0:43	7.54	4		10007	0	-1	6	0.9010989	0		10	2.3389795
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:59	1:04
20. 05. 2019 0:54	20. 05. 2019 7:44	01. 06. 2019 1:35	6.83	3	#stress_2x #alcohol_2x #med #food_3x #love_2x	10007	0	-1	7	0.5609756	-5		10	8.708365
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	0:55	1:00
19. 05. 2019 0:50	19. 05. 2019 8:09	20. 05. 2019 8:48	7.31	2.5	#fullmoon	10007	0	-1	6	0.6931818	0		10	5.9269724
From	To	Sched	Hours	Rating	Comment	Framerate	Snore	Noise	Cycles	DeepSleep	LenAdjust	Geo	23:56	0:01
17. 05. 2019 23:51	18. 05. 2019 7:22	30. 05. 2019 0:31	7.51	0		10007	0	-1	3	0.21428572	0		10	2.46947

Methodologies

1.Statistics based(we’ll set a threshold for the reading and assign sleep/awake)

2.ML(obtain features and set parameters to assign states for testing data)

3.DL(Same as ML but will be able to model much more complex patterns)

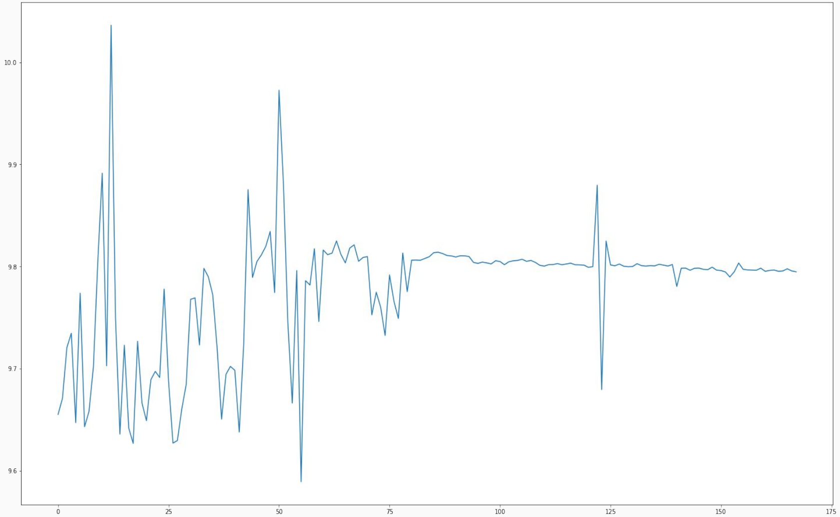
Data requirement increases from 1 to 3 . No. of users is as important as no. of days data we have.

Statistics based

We’ll set a threshold which will be based on the data from sleep as android so thst we can assign state to every 5 min window

To determine sleep start and end time we used something like a bandpass filter and see from what window we can have continuous

Basically we’ve used the fact that the readings will be nearly the same during sleep

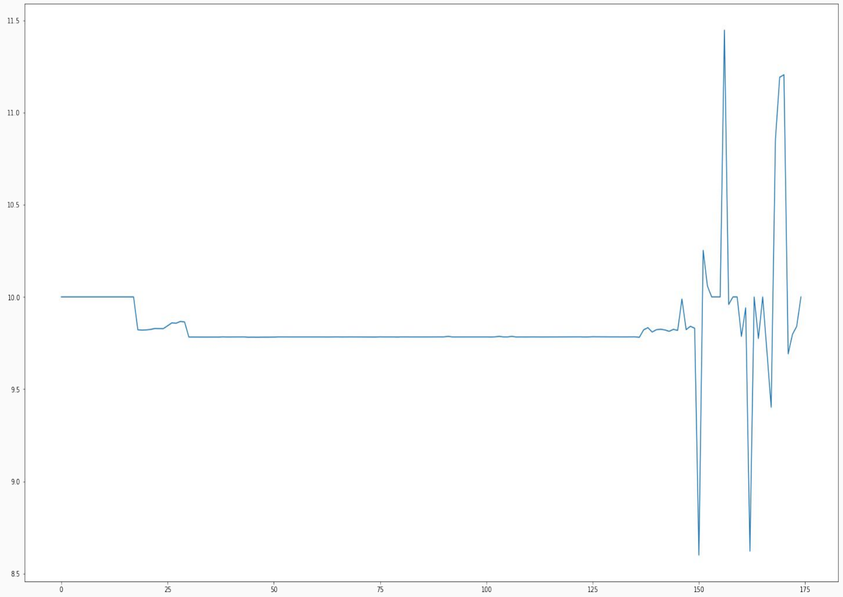


Using doze mode to our advantage

Because of Doze mode service is running but we’re not getting data during that period which is sometimes more than 2 hrs

We may use setallowwhileidle( )method but even using that method alarm fires only once for every 9 min

Doze mode implies device is stationary and screen is off so we stored a small while in the case of missing data but service is running



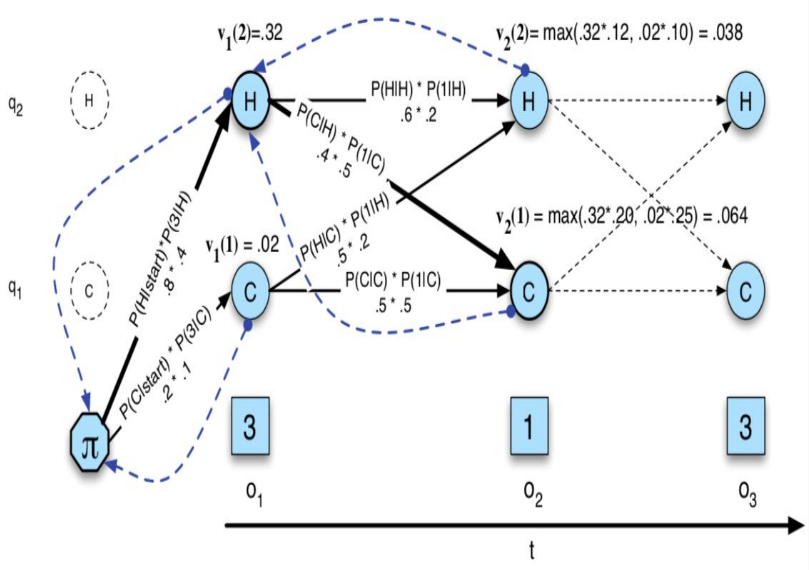
HMM

Markov model : Future states depend only on current but none of the past states.

We’ve used order 1 markov model ,i.e, next state depends only on current state

We used bauman walch for training the parameters(transition,emission probabilities)

After getting the parameters we used viterbi algorithm to get the best state sequence for the given sequence of observations



Cross lingual sentimental analysis

One major problem using HMM's to model sleep is the transition probability remains the same throughout but that's not the case in reality(probability that sleep->awake is low during night but high during morning)

Hindi doesn't have much of resources to do sentimental analysis .so we've mapped hindi word embeddings to english using regression and then used logistic regression to classify as positive or negative

We can do the same thing in our case ,i.e map our app data to sleep as android data using regression and then classify each window as sleep or awake using logistic regression but we don't have a good correlation between both the datas.

Once we know why their data is ranging from 1 - 10 and how are they assigning we can implement this

LSTM

LSTM’s have memory cells in which they store output of all previous observations in the time sequence in form of activation.

We can use bi-directional LSTM’s so that we see the patterns not only in a single direction but in both the ways

This procedure requires a lot of users as well as lot of data from each user and we should also have different type of users covering all the possibilities(early bird, night owl, who wakes frequently during night, who doesn’t wake at all) coz the model needs to see all possibilities to model them

Points to notice

We’re dealing with class imbalanced dataset ,i.e , fo eg take data of a random night and what you’ll notice is most of the(85-95%) of the windows will be sleep and only remaining (5-15%) will be sleep . In such a case we shouldn’t consider accuracy as the evaluation metric coz let’s say out of 100 windows you’ve actually were awake for 5 windows and if your algo isn’t able to detect those 5 as awake accuracy would be 95% but you didn’t even detect a single awake window. That’s the same thing happened in our case. We’re getting a good accuracy but data is not enough to detect awake windows. We should use Precision and Recall in case of class imbalanced data sets . The research paper we followed they mentioned only about accuracy but didn’t speak anything about precision and recall