Report of "Modeling People Count in a Building With Hidden Markov Model"

CalIt2 Data Set

The CalIt2 data set (Dua and Graff, 2019) comes in form of 48 time slices of 30-minute intervals for 15 weeks. The first measurement starts from July 24th, 2005, at time 00:00:00. From then on, every measurement shows either the number of people who enters the CalIt2 building or the number of people who exit it through the main door, within past 30 minutes. In order to simplify the data, the numbers of incoming and outgoing people were summed together for each day. Figure 1 shows the reprocessed data set, with the number of people passing through the door (regardless of whether they are entering or exiting the building) on the y-axis and the days passed since the first measurement (July 24th, 2005) in the x-axis.

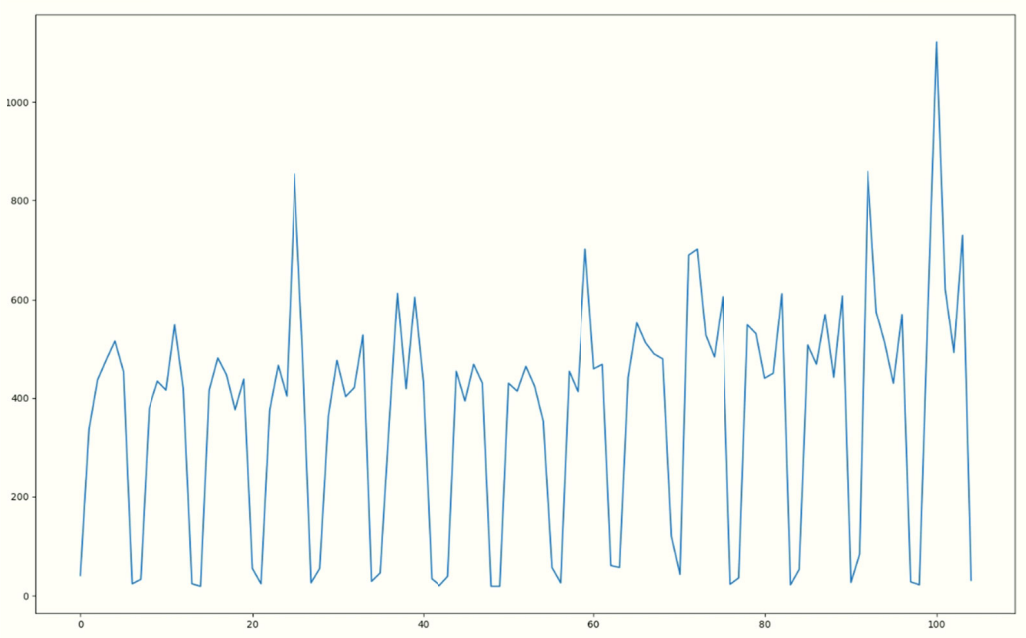

Figure 1.

Also given with the CalIt2 data set was the record of events that occurred within the building, such as conferences. Every event has start time and end time. In Figure 2, the existence of event was plotted against each day, where the y-value of a point is 1 if there was an event on that day, or 0 otherwise. It should be noted that if one is to run the source code of this project on the data set, the data files `CalIt2.data` and `CalIt2.events` occasionally contain blank lines that must be removed before running the code. I removed them manually.

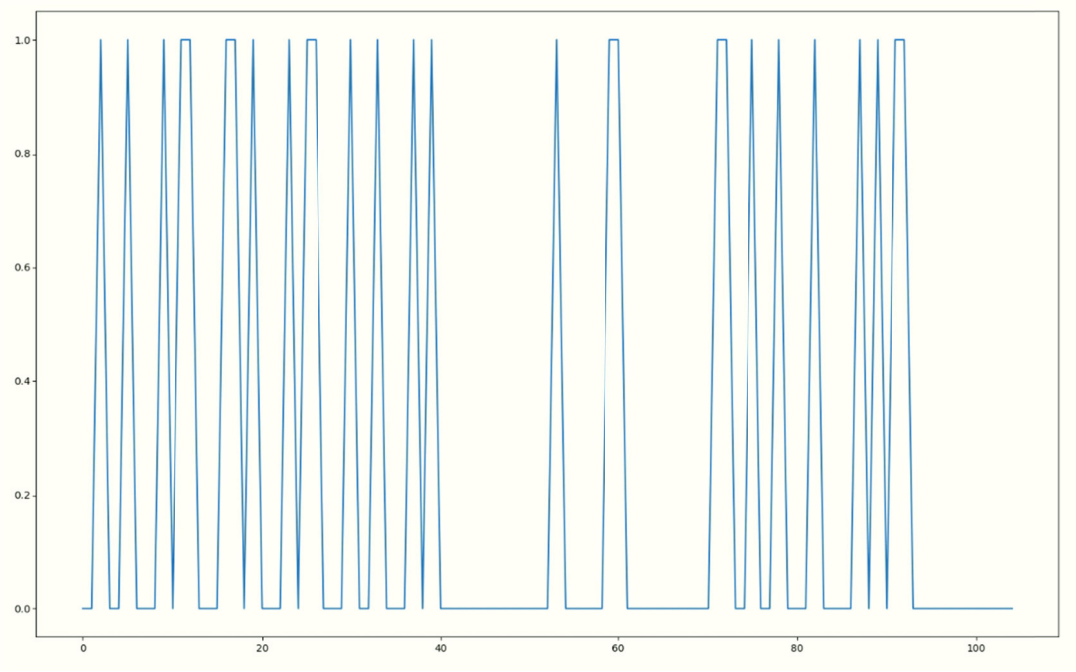

Figure 2.

States Configurations for HMM

To more closely discern the correlation between events and people count, Figure 3 was drawn such that the green plot indicates peak at dates with events, and blue plot is the people count data. From this figure, we can see that there are slight peaks in the people count when there are events on the same days, as is particularly evident around day 80. In between the events, often there are local minima. Since the events seem to correlate with people count, these events should be counted into the hidden state of the HMM to be designed. It is also evident that there are troughs corresponding to every weekend day. Thus, it is also expected that the hidden state contains the day of the week. Intuitively, there are to be fewer people entering or exiting the building in the weekends than in the weekdays.

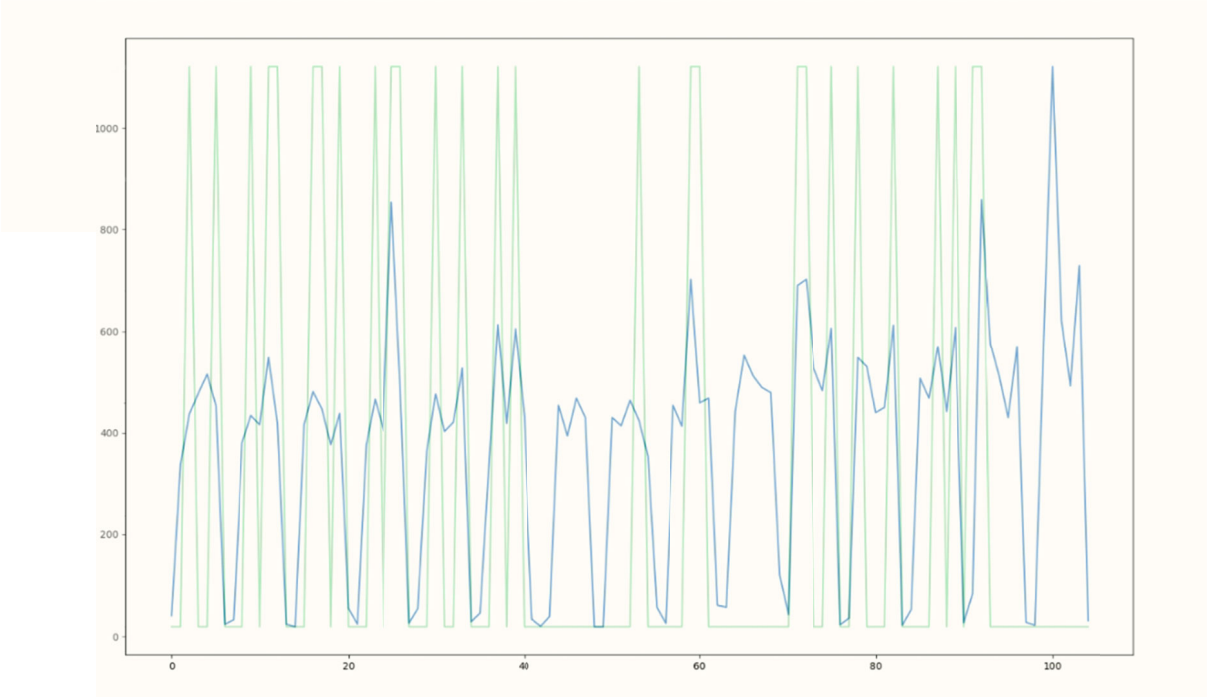

Figure 3.

The evidence variable is, obviously, the number of people observed to be passing through the main door of the building on a particular date. However, since there were often hundreds of people (and even over a thousand people at maximum) observed, the number of people itself could not be used as evidence space, not only because of inefficient memory usage, but also because of the possibility of overfitting. Thus, the number of people was divided into `num_intervals` number of uniform segments from 0 to maximum value, 1120. For the submitted code, `num_intervals` is set to 20.

Configuring the state variable was tricky. At first, the possible values for events were `"no_event"` and `"event"`, and the possible values for days of week were from 0 (Monday) to 6 (Sunday), so there were 14 states at first. However, this turned out to be insufficient, probably because as Figure 3 shows, events are often close to one another. Figuring that the information about events would have to be in longer series, the states were augmented such that the possible values for events were a variable number, demonstrated to be from 5 to 150 in the results section.

Initializing Probabilities

Initially, the start probabilities, transition probabilities, and emission probabilities were all given uniform probabilities. However, as a result, the model could not distinguish between the observations corresponding to weekdays with events and those corresponding to weekends without events, which had to be quite different intuitively. To break the symmetry, the probabilities were randomly initialized, with normalization. The problem of probabilities not matching the intuition persisted; the model would sometimes assign higher emission probability to higher people count on weekends, which is against our intuition that fewer

people would access the building on weekends. In both the case of uniform probabilities and the case of random probabilities, the Viterbi algorithm returned a sequence with log likelihood of only up to -140.38239109750424 on the training sequence.

Therefore, the probabilities were initialized intuitively. First, for the start probabilities, a high probability (10.0) was assigned to state corresponding to 0^{th} event value (which corresponded to "no event" before augmentation) and Sunday, and a very low probability (0.0000001) was assigned to the rest. This exactly matches our data set, which starts the measurement on Sunday, July 24^{th}, 2005, when there was no event. Second, for the emission probabilities corresponding to weekdays, a somewhat low probability (0.1) was assigned to observations corresponding to low people counts, and a higher probability (5.0) was assigned to the rest of the observations. For the emission probabilities corresponding to the weekends, a high probability (10.0) was assigned to the observations corresponding to low people counts, and a lower probability (0.1) was assigned to the rest of the observations. To determine the fraction of the observation values that would be considered as "low people counts," a hyperparameter `fraction` was declared, such that the lower $1/$`fraction` of observation values would be considered "low." For the submitted code, `fraction` is set to 4, such that the lower quarter of the observation segments is considered.

Lastly, the transition probabilities utilize a few different intuitions. A very low probability (0.0000001) was assigned to state transitions that do not follow the correct order of the days of week. For example, it is deemed nearly impossible to make a transition from Sunday to Sunday, or from Sunday to Saturday; significant probabilities are assigned only on transitions from Sunday to Monday. Also, the event values are supposed to follow the sequential order; the event value of i is likely to be followed by the event value of either i or $i+1$ (assuming that $i+1$ does not exceed the allowed index). All probabilities are normalized.

External Libraries Used with Changes

The implementation of HMM was brought from a third-party Python library called `hidden_markov` (https://hidden-markov.readthedocs.io/en/latest/index.html#). Because a line in the source code of the library had to be modified, the source code was included with the source code for this project. Within the method `train_hmm` of class `hmm`, the following line

```
emProbNew = emProbNew/ em_norm.transpose()
```

was changed to

```
emProbNew = emProbNew/ em_norm
```

Results and Discussion

Many machine learning tasks require prediction of original hidden states and ask for accuracy. This approach did not seem applicable for this case, because while the task would have been

to predict whether there was an event on a given date, the model used in this project incorporates event values whose meanings are difficult to interpret. For example, if 50 event values were used, which of these should be counted as "no event"? Since I could not find an answer to this question, another approach to quantitative measurement was adopted. The most likely sequence of hidden states was determined by Viterbi algorithm. Then, for each of the state in the sequence, the mean of distribution of observations was calculated based on its emission probabilities. This would result in a reconstructed sequence of observations. The mean squared error (MSE) and the mean of absolute error (MAE) ($\frac{1}{n}\sum_i|y_{actual} - y_{rec}|$) were calculated between the reconstructed observation sequence and the actual observation sequence. The log likelihoods of the most likely sequences from Viterbi algorithm were also computed. The result is as below, for different numbers of values for events.

num_events	train_mse	train_mae	train_log_prob	test_mse	test_mae	test_log_prob
5	8386.33	54.56	-81.71	43764.85	132.30	-429.67
10	6865.91	52.58	-72.50	42798.49	132.17	-410.17
20	3046.43	41.44	-37.72	39485.26	115.41	-395.48
30	1326.34	28.42	-17.81	22118.98	80.43	-392.61
40	948.54	27.22	-5.46	7574.57	63.94	-404.97
50	954.90	27.32	-5.43	15023.40	71.32	-381.75
60	956.36	27.34	-5.43	14542.43	80.70	-334.77
70	957.98	27.35	-5.43	13873.94	78.25	-328.73
80	959.59	27.37	-5.44	13661.48	78.03	-327.33
90	949.31	27.20	-5.45	13468.64	77.78	-326.29
100	948.50	27.23	-5.46	13292.09	77.49	-325.44
110	950.02	27.25	-5.47	13129.40	77.19	-324.70
120	938.34	27.10	-5.48	12978.73	76.87	-324.04
130	935.61	26.96	-5.50	12563.76	76.76	-323.44
140	916.67	26.70	-5.51	12396.88	76.51	-322.88
150	917.67	26.72	-5.53	12242.19	76.24	-322.35

It is worth noting how the huge drop in MSE occurs from 30 to 40 event values. Both the MSE and MAE values keep decreasing as more event values are used. Experiments could have been further conducted to find where the errors would plateau.

Figure 4 and Figure 5 compare between the actual observation sequence and the reconstructed ("Predicted") observation sequence, on training sequence and the test sequence respectively. The reconstructed data closely match the actual observation sequence on training sequence. Also, qualitatively, the reconstructed data seem to highly correlate with the actual data. Nevertheless, the MSE for test sequence is very high, which suggests that the particular Hidden Markov Model does not generalize well to unseen data.

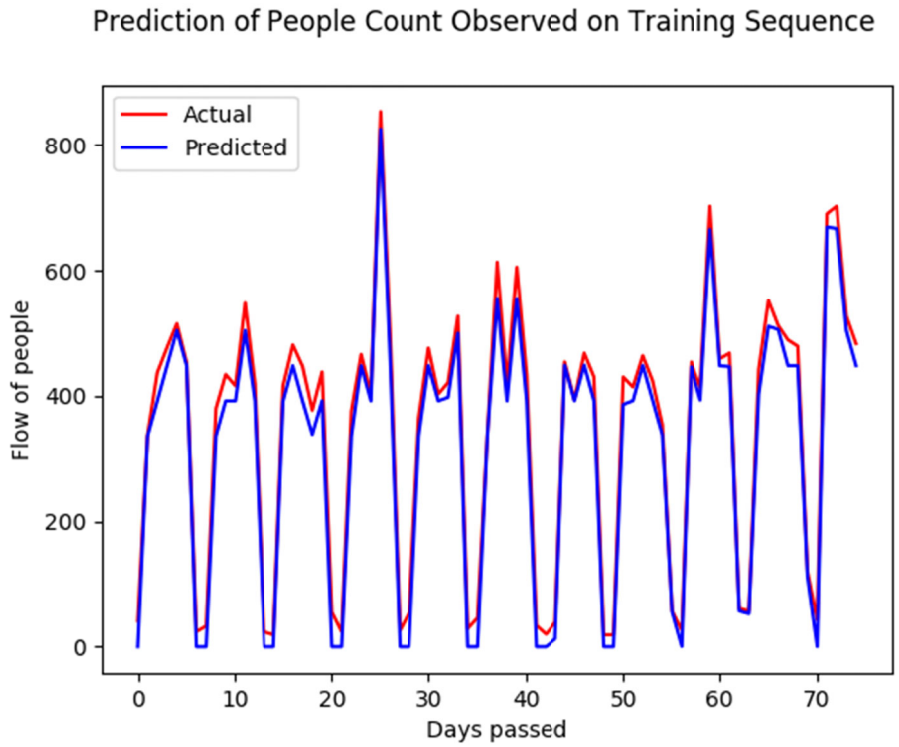

Figure 4.

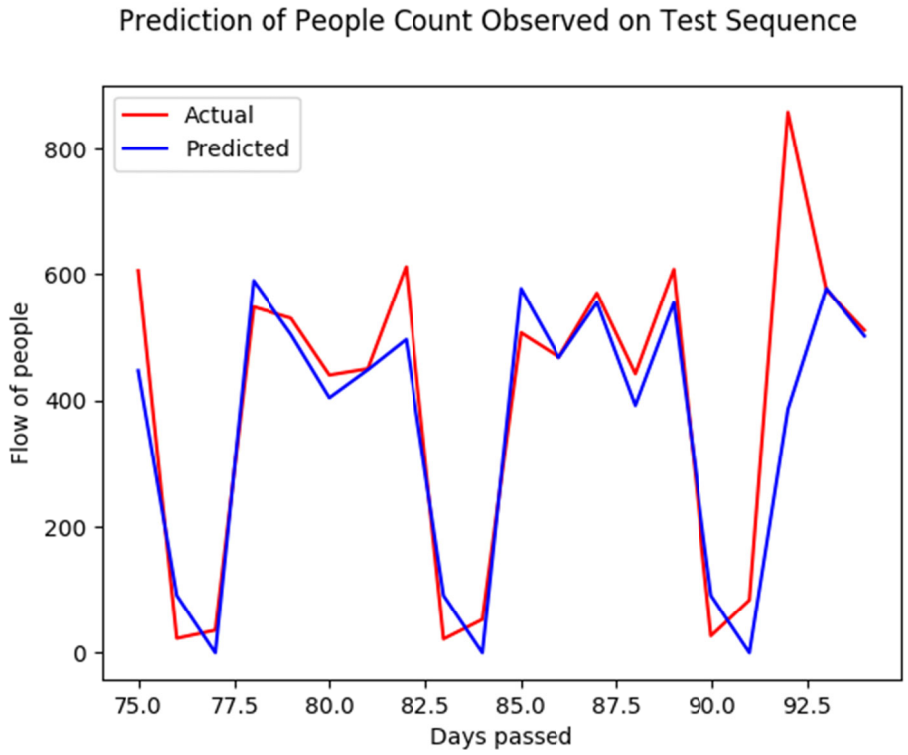

Figure 5.

Reference

Dua, D. and Graff, C. (2019). *UCI Machine Learning Repository*

[http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.