Using Markov Models for Covid Prediction [*]

Janaan Lake

May 5, 2022

1 Introduction

Many machine-learning and prediction problems involve data sampled over time. In time series problems, observations close to each other in time are expected to be more similar than observations far away. A variety of applications use time-series data including weather forecasting, stock price prediction, forecasting unemployment or interest rates, predicting server utilization by hour, etc. Much of the data regarding Covid-19, including infection rates, active cases and deaths are an example of time-series data. For the class project, I wanted to use data related to the recent event of the pandemic and to apply a probabilistic model to the time-series prediction problem.

2 Models

Markov and Hidden Markov Models are engineered to handle data which can be represented as a 'sequence' of observations over time, which makes them a candidate for time-series prediction models.

2.1 Markov Chain

Markov chains are a fairly common and relatively simple way to statistically model random processes. A Markov chain essentially consists of a set of transitions, which are determined by some probability distribution that satisfy the Markov property. The Markov property is the assumption that the current state only depends on a finite fixed number of previous states. For example, a first-order Markov chain is a model where the current state depends only on the previous state and not on any earlier states. Therefore, the transition model is the conditional distribution

$$P(\mathbf{X}_t|\mathbf{X}_{0:t}) = P(\mathbf{X}_t|\mathbf{X}_{t-1}).$$

The transition model for a second-order Markov chain is the conditional distribution

$$P(\mathbf{X}_t|\mathbf{X}_{0:t}) = P(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1}).$$

The Markov Chain also utilizes the stationary process assumption, which is the assumption that the conditional probability distribution over the next state, given the current state, doesn't change over time.

[*] CS 6190; Spring 2022 Instructor: Dr. Shandian Zhe, University of Utah

Markov chains are easy to implement and to learn. However, the Markov assumption typically makes these models unable to successfully produce sequences in which some underlying trend would be expected to occur. They therefore lack the ability to produce context-dependent content since they cannot take into account the full chain of prior states.

2.2 Hidden Markov Model

Hidden Markov models are probabilistic frameworks where the observed data are modeled as a series of outputs generated by one of several (hidden) internal states. HMMs can be used as a black-box density models on sequences. They have the advantage over Markov models in that they can represent long-range dependencies between observations, facilitated via the latent or hidden states. Hidden Markov Models use the Markov assumption and the stationary process assumption. The transition model for the hidden states is similar to the conditional probability distribution described above for the Markov chain.

The HMM model also uses the output independence assumption, which is that the output observation is conditionally independent of all other hidden states and all other observations when given the current hidden state. The observation model is the probability of the observation (or emission) given the current hidden state $\mathbf{X}_i$:

$$P(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = P(\mathbf{E}_t|\mathbf{X}_t).$$

In addition to the transition and observation models, a HMM contains an initial state model, which is the prior probability distribution at time 0, $P(X_0)$. Given these models, we can express our HMM with the complete joint distribution over all the variables:

$$P(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = P(\mathbf{X}_0)\prod_{i=1}^{t} P(\mathbf{X}_i|\mathbf{X}_{i-1})P(\mathbf{E}_i|\mathbf{X}_i).$$

The transition and observation models can be learned from observations using the Baum-Welch algorithm, which is a special case of the expecation-maximization, or EM algorithm. The inference task is done using the Viterbi algorithm, which is a recursive algorithm that computes the posterior distribution over the future state, given all the evidence to date.

3 Data

This project used Covid-19 data from the state of Utah. The data was downloaded from the Utah coronavirus dashboard on February 25, 2022 and contains data from March 18, 2020 through February 24, 2002[1]. The data consisted of 722 samples and contained daily reported Covid-19 case counts, total active case counts and daily death counts in the state of Utah.

3.1 Case Counts

The models were initially created to predict changes in certain metrics from day to day. The were three types of metrics modelled and predicted in the project: daily reported case counts, total active case counts, and daily deaths. The change in each of these metrics was the quantity predicted. The range of predictions of these models fell within the range of numbers representing the difference between the maximum decrease and the maximum increase of the training data. For example, the change in daily reported case counts had a maximum decrease of $-3,757$ and a maximum increase of $3,789$. Therefore, the range for this metric was $7,546$. The emissions for this model was an integer value within this range. The other metrics were treated similarly.

3.2 Labels

Another approach used for predicting Covid-19 information was to label the Covid data based on the rate of change, the overall number of cases and the direction of the change (increase/decrease). The idea was borrowed from a paper that used HMMs for Covid-19 prediction in India [2]. For each metric (daily reported cases, total active cases, and daily death counts), a label was assigned to the overall number. The set of labels for the current counts is {L,M,H,E}, representing a level of (L)ow, (M)edim, (H)igh, and (E)xtreme for the daily count. These labels were assigned by first calculating the standard deviation of the metric for the entire dataset and then comparing the daily count to its relative distance from 0. For example, the daily reported cases ranged from 0 to $13,524$ for the entire dataset, with a standard deviation of $1,771$. If the daily reported case count fell between 0 and 1/2 of the standard deviation, then the case count was considered “Low” and labeled “L”. The ranges for the labels are listed in Table 1, with σ representing the standard deviation.

Range	Label
$0 \leq x < \frac{1}{2}\sigma$	”L”
$\frac{1}{2}\sigma \leq x < \sigma$	”M”
$\sigma \leq x < 2\sigma$	”H”
$2\sigma \leq x$	”E”

Table 1: Ranges for the Current Counts Labels

A similar process was used for labeling the daily change in the counts, but only three labels were used: {(L)ow, (M)edium, and (H)igh}. The ranges for determining the labels for the daily change in counts for each metric are listed in Table 2.

Range	Label
$0 \leq x < \frac{1}{2}\sigma$	”L”
$\frac{1}{2}\sigma \leq x < \sigma$	”M”
$\sigma \leq x$	”H”

Table 2: Ranges for the Daily Change Labels

Lastly, if the change in counts was an increase, then the label was appended with a “+”. A decrease in the change was labelled with a “-“. As an example, if a certain day had the label “HM+” this indicated a “High” level of daily case counts, a “Medium” rate of change from the prior day and an increase in the change.

4 Experiments and Results

The Markov chain model as well as the HMM model were used to predict case counts and labels for ten consecutive days. For the HMM, four hidden states were used because that empirically produced the best results. A baseline was established for comparison for each data type that was predicted.

4.1 Measuring Performance

The performance of the models for predicting changes in case counts was evaluated using the average difference in the test data between the actual change and predicted change. A score of 0 would indicate a perfect prediction, with performance accuracy decreasing as the difference increased.

The performance metric of the models for predicting labels is a bit more complex. Each individual component of the predicted label is compared to the actual label in the test data, and the total accuracy measure is essentially a weighted score of each of these components. For instance, the daily label is compared to the actual label and weighted. The relative difference between the labels is considered in the scoring. For example, if both the predicted daily score and the actual daily score are H (indicating a "High" number of counts), then that portion of the weighted score is 100%. If the difference is one level, (i.e. H and M, M and L, etc.) then the score is 75%. If the difference is two levels, the score is 50% and 25% for a difference of 3 levels. The change label is treated similarly. Finally, the sign of the change is compared. If both are positive, indicating an increase, that portion of the weighted score is 100%. Otherwise, it is 0. While this score may not be a perfect measure of accuracy, it is the comparison between models using the same score that is most important and informative.

4.2 Results

Each type of data (numerical vs. labeled) was evaluated using a baseline. The baseline consisted of calculating the probability of the feature occurring in the training data. For prediction, each day was then sampled from that probability distribution. Then each data type was modelled using a first-order Markov chain and a Hidden Markov Model. The labelled data was additionally evaluated on a second-order Markov chain model. The numerical data could not be modelled on a second-order Markov chain because the input space was too large. Additionally, for each data type three types of metrics were used: daily reported covid cases, total active cases, and daily death counts.

The results of the numerical data, i.e. the daily change in counts, are reported below in Table 1. This table shows the average difference between the predicted and actual changes in the daily counts. As a reference and to provide further information, Table 2 shows the difference as an amount of the standard deviation of change for each metric to give a relative measure of comparison across metrics. For example, the baseline measurement for the reported cases metric is $3,433$, which represents an average difference between the predicted daily change and the actual change of $3,433$! This amount is 6.2 standard deviations for this metric.

Metric	Baseline	1st-Order MC	HMM
Change in Reported Cases	3,433	1,564	199
Change in Total Estimated Active Cases	6,573	3,812	252
Change in Deaths	12	2	2

Table 3: Average Difference between Predicted and Actual Changes for Numerical Data

Metric	Baseline	1st-Order MC	HMM
Change in Reported Cases	6.2	2.8	0.4
Change in Total Estimated Active Cases	3.3	1.9	0.1
Change in Deaths	3	0.5	0.5

Table 4: Number of Standard Deviations of Average Difference between Predicted and Actual Changes for Numerical Data

The results of the labelled data are reported in Table 3. The accuracy measure described in Section 4.1 was used.

Metric	Baseline	1st-Order MC	2nd-Order MC	HMM
Change in Reported Cases	75.2%	87.1%	87.9%	87.6%
Change in Total Estimated Active Cases	73.5%	92.4%	93.0%	90.3%
Change in Deaths	71.5%	80.5%	79.1%	81.6%

Table 5: Accuracy of Predicted Labels

5 Analysis

For the numerical data, the 1st-Order Markov Chain offered an improvement over the baselines score, indicating that the model was able to find some predictive patterns in the training data. The Hidden Markov Model offered even more improvement for the change in daily case counts and total active cases. This indicates further refinement of learning underlying context-specific trends that is facilitated by the latent variables. For example, the measured difference in change for the daily reported case counts metric decreased by more than half using the 1st-Order Markov Chain. Using the HMM, the score improved nearly 7 times! However, this improvement is somewhat inflated only because the baseline score was very high due to the sparsity of data points in the training set within the range of outputs.

For the labelled data, the 1st and 2nd-Order Markov Chain and the HMM models show improvement from the baseline scores for all metrics, but each model's performance does not differ significantly from the other model's performance. This indicates that the learning on the training data found a Markov process that captured some of the underlying patterns and trends. However, the 2nd-Order and HMM models were not able to refine the learning process. Some of the difficulty is due to the fact of a smaller dataset that only contained 722 samples over a relatively short amount of time. The dataset also contained somewhat random-looking spikes that further complicated learning underlying patterns and trends. Also, the labelled data involved a much smaller input and output space than the numerical data. Therefore, the baseline score for the labelled data was more accurate and hence offered less opportunity for improvement.

6 Future Plans

As just mentioned, the dataset was small. Future work could involve testing these models on Covid-19 datasets that contain many more samples and a longer time period. Would the performance improve and how would the results of the numerical data then compare to the labelled data? Additionally, I would like to try a different learning algorithm for the HMM. If the HMMs are used for time series prediction, more sophisticated Bayesian inference methods, like Markov chain Monte Carlo (MCMC) sampling are proven to be favorable over finding a single maximum likelihood model both in terms of accuracy and stability [3].

Please refer to the following github repository for the project notebook:
https://github.com/JanaanL/Covid_19_Project.git

References

[1] Utah coronavirus dashboard.

[2] Shreekanth Prabhu and Natarajan Subramanyam. Surveillance of covid-19 pandemic using hidden markov model. 08 2020.

[3] I. Róbert Sipos. Parallel stratified mcmc sampling of ar-hmms for stochastic time series prediction. *Proceedings, 4th Stochastic Modeling Techniques and Data Analysis International Conference with Demographics Workshop*, pages 295–306, 2016.