# STATS 620 Project

## Learning Discrete Time Markov Chains Under Concept Drift

Kevin Wibisono
University of Michigan, Ann Arbor

## 1 Introduction

Learning under concept drift is an important research area as it allows practitioners to use their models on non-stationary data generating processes. While there have been a number of works pertaining to this topic, Roveri's (2019) paper is the first one to focus on discrete time Markov chains (DTMCs). It introduces change-detection mechanisms assuming the Markov chains are homogeneous.

Consider a DTMC $\Theta = \{\pi, P(t)\}$, with $\pi$ being the initial distribution and $P(t)$ the transition matrix at time $t$. Let $t^*$ be a point at which $P(t) = P_0$ for every $t < t^*$ and $P(t) = P_1$ for every $t \geq t^*$. In other words, the concept drift occurs at time $t^*$. Moreover, assume we have access to the first $L < t^*$ observations. In order to detect the concept drift, this paper introduces three algorithms. We focus on two of them, namely P-CDM and NP-CDM. Here, P and NP refer to parametric and non-parametric, respectively.

**Report organization** Section 2 explains their proposed P-CDM and NP-CDM algorithms. Section 3 performs in-depth simulation studies on the performance of the algorithms. Section 4 concludes this report.

**Code** Our code is available in our Github repository https://github.com/k-wib/mcdrift. In addition, results in Section 3 can be reproduced by running the notebooks provided in the repository.

## 2 Algorithms

Throughout this section, let $\mathcal{T} = \{s_1, s_2, \cdots, s_T\}$ be the observed sequence, where $s_i \in \{1, 2, \cdots, N\}$. Also, assume that all Markov chains are irreducible, positive recurrent and aperiodic. Our goal is to estimate $t^*$, the time corresponding to the first concept drift occurrence.

### 2.1 P-CDM

In P-CDM, we assume that $P_0$ and $P_1$ are known. Let $w_i = \{s_{W(i-1)+1}, \cdots, s_{Wi}\}$ be non-overlapping subsequences of length $W$. For each $i$, we can compute the log of the likelihood ratio $l_i = \log(\mathbb{P}_{\Theta_1}(w_i)/\mathbb{P}_{\Theta_0}(w_i))$. Here, $\mathbb{P}_{\Theta_k}(w_i)$ $(k \in \{0, 1\})$ refers to the probability of observing $w_i$ given the transition matrix $P_k$. By the Markov property, it is easy to see that

$$\mathbb{P}_{\Theta_k}(w_i) = \pi_k^{W(i-1)+1}\left(s_{W(i-1)+1}\right) \prod_{j=W(i-1)+1}^{Wi-1} p^k_{s_j, s_{j+1}}.$$

Here, $\pi_k^a(b)$ refers to the probability that a DTMC $(\pi, P_k)$ is in state $b$ at time $a$, and $p_{c,d}^k$ refers to the transition probability from state $c$ to $d$ of the DTMC $(\pi, P_k)$. In order to simplify the calculations, $\pi_k^a(b)$ is estimated by the asymptotic probability $\pi_k^{\text{asymp}}(b)$, which is assumed to exist. We denote the estimated log likelihood ratio by $\tilde{l}_i$.

The key idea of this algorithm is the fact that $l_i > 0$ ($l_i < 0$) implies it is more likely for $\Theta_1$ ($\Theta_0$) to generate $w_i$. Specifically, we start from $m_0 = 0$, and for each $i \geq 1$, we recursively calculate

$$m_i = \max\left(0, m_{i-1} + \text{sign}\left(\tilde{l}_i\right)\right).$$

We then detect a change in $w_\alpha$ (the $\alpha$-th subsequence) once $m_\alpha = K$, for some positive integer $K$.

From this formulation, we observe the following points:

1. The value of $K$ introduces a trade-off between wrong detection and detection delay. For example, consider the case where $K$ is very large. While the detection will most likely be correct, it can happen at time $t^{**}$ much larger than $t^*$.
2. The larger the subsequence length $W$, the less meaningful the detection will be. For eample, we prefer to say that a shift may occur between time 100 and 110, than between time 100 to 200.
3. The closer $P_0$ is to $P_1$, the more difficult it is to detect the shift.

### 2.2 NP-CDM

In NP-CDM, the transition matrices $P_0$ and $P_1$ are unknown. However, we assume that the first $L$ observations in the sequence $\mathcal{T}$ is generated under $(\pi, P_0)$, for a large enough $L$ which allows us to accurately estimate $P_0$ using classical MLE. In order to estimate $P_1$, we rely on $L$ most recently acquired observations. Similar to P-CDM, the initial probabilities are estimated by the asymptotic probabilities.

Concretely, we first estimate $(\pi_0, P_0)$, $(\pi_1, P_1)$ from $\{s_1, \cdots, s_L\}$. Same as before, we start from $m_0 = 0$, and recursively calculate

$$m_i = \max\left(0, m_{i-1} + \text{sign}\left(\tilde{l}_i\right)\right)$$

for each $i \geq 1$. After each update, we check if $m_i = K$. If yes, we detect a change and stop. Otherwise, we re-estimate $P_1$ using $\{s_{Wi-L+1}, \cdots, s_{Wi}\}$ when $Wi > L$. We then re-estimate $\pi_1$ using the asymptotic distribution corresponding to the new $P_1$.

Apart from the points we mentioned above, these observations are worth noting:

1. NP-CDM corresponds to a more realistic case where we only observe a sequence of states, without knowledge of the transition matrices.
2. The parameter $K$ is more important in NP-CDM since we now rely on estimates of $P_0$ and $P_1$. Specifically, the same value of $K$ can lead to a higher false positive rate.

# 3 Experiments and results

## 3.1 Average detection time for different values of $W$ and $K$

In this experiment, we compare the change detection performance of P-CDM and NP-CDM for different values of $W$ and $K$. Concretely, we set $N = 5$ (number of states), $T = 10,000$ (sequence length), $t^* = 2,000$ (time abrupt change occurs), $L = 1,000$ (number of observations guaranteed to come from $P_0$ and $\pi = (0.2, 0.2, 0.2, 0.2, 0.2)$ (initial distribution). We randomly generate each element of $P_0$ and $P_1$ independently from the standard uniform distribution, and normalize each row so that it sums up to 1.

When estimating the time of the abrupt change, we take the average of the interval in which the change is predicted to occur. For example, when $W = 10$ and the change is detected in $s_k$ with $91 \leq k \leq 100$, the estimated time is simply $(91 + 100)/2 = 95.5$. We repeat this experiment 500 times and report the average detection time for each $(W, K)$ pair. The results are shown in Tables 1 and 2 below.

Table 1: Average detection time for P-CDM

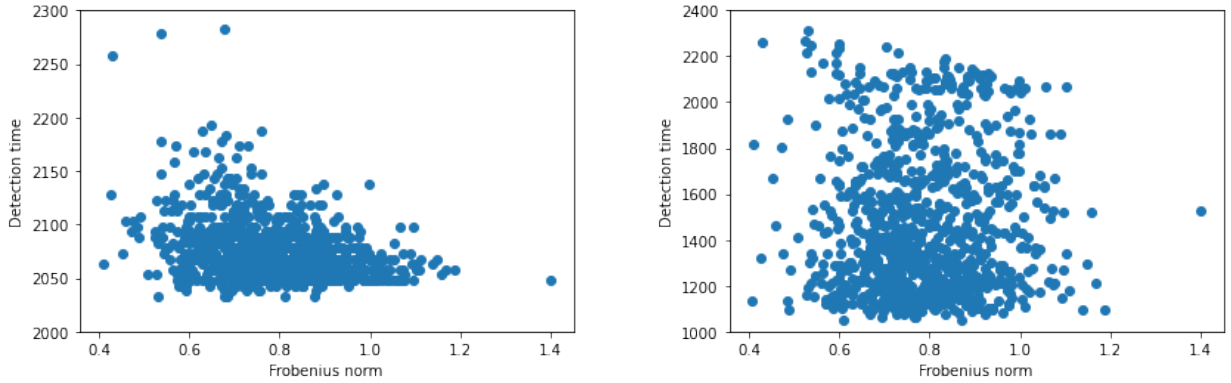
| **W \ K** | **1** | **2** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **2** | 6 | 28 | 590 | 1725 | 2105 | 2433 |
| **5** | 28 | 221 | 1858 | 2073 | 2155 | 2400 |
| **10** | 132 | 1106 | 2043 | 2116 | 2240 | 2614 |
| **50** | 1945 | 2074 | 2226 | 2477 | 2980 | 4489 |
| **100** | 2044 | 2150 | 2450 | 2950 | 3950 | 6951 |

Table 2: Average detection time for NP-CDM

| **W \ K** | **1** | **2** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **2** | 1005 | 1030 | 1132 | 1493 | 2202 | 3248 |
| **5** | 1048 | 1140 | 1527 | 2228 | 3414 | 4088 |
| **10** | 1096 | 1261 | 1874 | 2994 | 3910 | 4429 |
| **50** | 1459 | 1969 | 3432 | 4368 | 4779 | 5929 |
| **100** | 1842 | 2751 | 4170 | 4747 | 5515 | 7843 |

We observe that as $W$ and $K$ increase, the average detection time increases. In P-CDM, the optimal $K$ is 20 when $W = 2$, 10 when $W = 5$, 5 when $W = 10$, and 1 when $W = 50$ or 100. In NP-CDM, the optimal $K$ is 20 when $W = 2$, 10 when $W = 5$, 5 when $W = 10$, 2 when $W = 50$, and 1 when $W = 100$. Here, optimality is determined from the parameter which results in an average detection time that is the closest to $t^* = 2,000$.

Looking at Table 1, it seems that when the transition matrices are known, a large value of $W$ with a very small value of $K$ tends to perform very well. On the other hand, with a small value of $W$, we need to be more careful in picking a sufficiently large $K$ to avoid false early detection. In the more realistic case where the transition matrices are unknown, Table 2 suggests that we might need to further increase $W$ to achieve the same goal. However, as mentioned before, a large value of $W$ will make the detection less meaningful. If we are satisfied with taking the average of the interval to be our point estimate, we do not need to worry to much about this downside.




(a) P-CDM (b) NP-CDM

Figure 1: Detection time versus Frobenius norm

### 3.2 Average detection time versus the distance between $P_0$ and $P_1$

Earlier, we conjectured that shift detection is harder when $P_0$ and $P_1$ more similar to each other. In this subsection, we would like to empirically check if the conjecture is correct. For simplicity, we use the same hyperparameter settings as used in Section 3.1. Also, we select $(W, K) = (5, 10)$, a hyperparameter pair that works reasonably well for both methods as seen in Tables 1 and 2. Again, to estimate time of the abrupt change, we take the average of the interval in which the change is predicted to occur. We repeat this experiment $1,000$ times and record the estimated time as well as $||P_0 - P_1||$ for each repetition. The results are summarized in Figures 1 and 2 below.

Figure 1 seems to confirm our conjecture for P-CDM. In particular, as the distance between $P_0$ and $P_1$ increases, we tend to see detection times closer to $t^* = 2,000$. The same, however, cannot be said for NP-CDM. Also, one important result worth pointing out is that out of $1,000$ repetitions, NP-CDM detects no change in 73 of them (i.e., false negative), while there are no false negatives in P-CDM. This is perhaps not surprising considering we do not have access to $P_0$ and $P_1$.

### 3.3 False negative rates of P-CDM and NP-CDM

Results in Table 2 (and potentially Table 1) can be misleading due to the possible presence of false negatives for which we set a detection time of $T = 10,000$. In this subsection, we investigate the false negative rates of both algorithms, for the same parameters and different values of $W$ and $K$ as used in Section 3.1. For P-CDM, all $(W, K)$ combinations lead to a false negative rate of 0%, with the exception of $(W, K) = (2, 50)$ with a false negative rate of 1%. While false negatives occur sporadically for P-CDM, the same cannot be said for NP-CDM, as shown in Table 3.

Table 3: False negative rate for NP-CDM (in %)

| **W \ K** | **1** | **2** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **2** | 0 | 0 | 1 | 3 | 7 | 13 |
| **5** | 0 | 1 | 4 | 8 | 18 | 22 |
| **10** | 1 | 2 | 6 | 15 | 22 | 25 |
| **50** | 3 | 7 | 19 | 26 | 26 | 26 |
| **100** | 6 | 14 | 24 | 26 | 26 | 26 |

We observe that false negatives occur sporadically for P-CDM, yet very commonly for NP-CDM. From Tables 2 and 3, we can see that if minimizing the false negative rate is our utmost priority, we should choose small values of $W$ and $K$ at the expense of possible early detections.

### 3.4 False positive rates of P-CDM and NP-CDM

We now turn our attention to false positives, which happen when the algorithm detects a change when such a change never occurs. We use the same parameters as in Section 3.1, with the exception that the transition matrix $P_0$ is used throughout. Tables 4 and 5 show the false positive rates for P-CDM and NP-CDM.

Table 4: False positive rate for P-CDM (in %)

| **W \ K** | **1** | **2** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **2** | 100 | 100 | 97 | 49 | 7 | 1 |
| **5** | 100 | 100 | 46 | 1 | 0 | 0 |
| **10** | 100 | 95 | 3 | 0 | 0 | 0 |
| **50** | 26 | 1 | 0 | 0 | 0 | 0 |
| **100** | 1 | 0 | 0 | 0 | 0 | 0 |

Table 5: False positive rate for NP-CDM (in %)

| **W \ K** | **1** | **2** | **5** | **10** | **20** | **50** |
|---|---|---|---|---|---|---|
| **2** | 100 | 100 | 100 | 97 | 94 | 84 |
| **5** | 99 | 99 | 96 | 91 | 84 | 51 |
| **10** | 99 | 97 | 93 | 87 | 71 | 24 |
| **50** | 96 | 92 | 84 | 60 | 22 | 1 |
| **100** | 92 | 88 | 74 | 39 | 8 | 0 |

These results verify the conjecture we made earlier that the same value of $K$ can result in a higher false positive rate in NP-CDM. In general, we see higher false positive rates for small $W$ and $K$ values.

## 4 Conclusion

From our experiment results, we have the following conclusions. First, the choices of $W$ and $K$ are extremely crucial and can lead to very different behaviors in both algorithms; domain knowledge about specific use cases of the algorithms might be helpful in this case. Second, for P-CDM, it seems like a large value of $W$ and a small value of $K$ are desirable as they correspond to an accurate detection (on average) and small false positive and false negative rates. The situation, however, is much more complicated for NP-CDM.

## 5 References

M. Roveri, "Learning Discrete Time Markov Chains Under Concept Drift," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9, pp. 2570-2582, Sept. 2019, doi: 10.1109/TNNLS.2018.2886956.