Predictive Policing of Vancouver Crime with Hidden Markov Models

Cody Griffith
Department of Mathematics
The University of British Columbia
cgriff@math.ubc.ca

Ziming Yin
Department of Mathematics
The University of British Columbia
ziming@math.ubc.ca

Tim Jaschek
Department of Mathematics
The University of British Columbia
jaschekt@math.ubc.ca

Abstract

The aim of this work is to predict time and location of possible future crimes with unsupervised machine learning techniques. We generate a time-chain of GPS locations of likely crimes in the city of Vancouver with a hidden Markov model (HMM). As a training set we use data set consisting of different classes of crimes with time and location. The dataset contains more than half a million of entries and is publicly available. Unfortunately We did not find crucial hidden states that would improve the prediction. However, a different initialization of our model might provide significantly better results.

1 Introduction

The path of predictive policing has taken many twists and turns in attempting to use ever increasingly large datasets to better allocate police resources. There are many fantastic examples of this being done successfully to help minimize local crime and agencies world wide have taken notice. An interesting model, called "*Series Finder*" [13], has the ability to detect crime patterns purely from data and has helped trained professionals to update their own database while also outperforming the competitors. However, this algorithm has not been applied on large scales yet.

In practice, most of the available predictive policing tools are concerned with predicting areas where crimes are most likely to happen. Although big companies like IBM offer predictive policing tools, one of the most widely deployed products comes from a small Santa Cruz, Calif. firm called "*PredPol*" [11]. According to the company's website their algorithm is based on machine-learning and is cities including Los Angeles, Chicago and Atlanta, but due to licensing and proprietary issues, we aren't sure what techniques are being done.

Here in Vancouver, the first city in Canada to attempt predictive policing, a software called "*GeoDash*" [4] gives real-time updates to patrolling officers for when and where crimes have occurred for nearby homes. This software is more than just an accessible database though, as it has the ability to also allocate officer time by forming a heat map of for future break-ins with probabilistic methods. This of course leads to prevented crime as well as optimizing police funding and public safety. Even though they do not discuss machine learning aspects, we would like to point out the works of [8] and [3], which provide helpful visualizations of crime data in Vancouver.

More examples for the applications of machine learning to crime data can be found in [9], where linear regression, additive regression and decision stumps are used, in [7], where a deep neural network is

Final project for CPSC540 (The University of British Columbia 2018), Vancouver, B.C., Canada.

fitted to crime data and [1], whoses authors follow a similar strategy to ours by constructing a hidden Markov model.

Although, there is never good without bad and there have been failures of predictive policing. The most famous example was in Los Angeles where a model had begun to be used but it had been trained on racially profiled data. Thus, the model began to racially profile and the department had the luxury of pointing the finger at a computer when a minority was treated with injustice. Aspects of biased arrests are discussed in the paper [2]. What must be taken from this is not that Machine Learning and Policing should not be mixed, rather that this area is extremely sensitive and models need to be built on data that is pure in intent. For the purposes of all models mentioned above, *Series Finder*, *GeoDash* and the tools of *PredPol*, human specific data was not used in any point of the learning or training of the model. It is here that we take inspiration and apply this to our own model.

We were unable to find algorithm-specific approaches to the *GeoDash* model for proprietary reasons, this also is a much more cumbersome project with entire data warehouses and multi-stream project pipelines that would be impossible for three poor graduate students to replicate. Instead, we consider applying a Hidden Markov Model (HMM) to open source data that can be found at [5] with the aim to both forecast future crime and isolate location of crime within Vancouver, British Columbia. We choose to use an HMM as we are not criminal analysts and have no keen eye for spotting patterns within data, instead we hope to train an HMM to find hidden clusters and interdependences for us to better predict the future.

2 Exploratory Data Analysis

The dataset we used is publicly available and weekly updated by the Vancouver Police Department [5]. It is roughly 30MB in size and contains information on the type of crime (i.e theft, breaking and entering, etc.), time (i.e year, day, hour and minute) and the coordinate of crime (i.e X and Y). This dataset goes as far back as 2003 and has a total of over 500,000 entries. From a simple exploratory search seen in Figure 1, we consider the histogram of frequencies across the various time scales. The following conclusions are made;

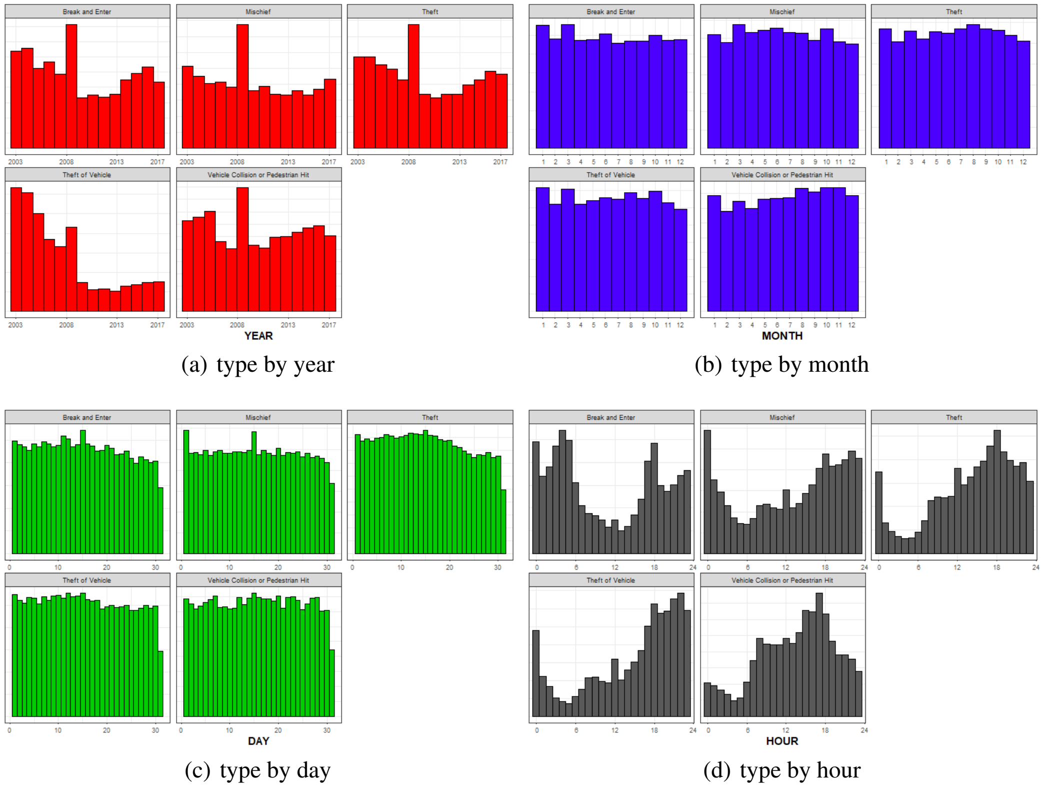
- Across years in Figure 1(a) we notice here that there is a spike in almost all types of crime for 2008 and following that a steady decrease in most until another rise in 2013. We do not know the reason for the peak in 2008. Surprisingly neither the Vancouver Olympic Games 2010 nor the Vancouver Stanley Cup riots from 2011 had a significant effect on the number of crimes committed in the year.
- Across months in Figure 1(b) all types of crime seem to be near constant for each month.
- Across days in Figure 1(c) we have that day 31 doesn't happen for ever month so it is noticeably lower in count. Small spikes seem to occur near the middle of the month, but everything is rather uniform.
- Across hours in Figure 1(d) we find the most interesting behavior. All types seems to behavior rather erratically with the general trend that crime seems to happen either early morning or late at night.

From our analysis, it becomes clear that we should consider to model at the hour scale as most other time scales are close to uniform in nature and thereby would not result in an interesting model. The hour scale on the other hand is best in two key ways: there is clearly some behavior that varies between crimes and across hours to present a complex interaction and there is realistic and practical value in having a model of crime on the hour. This could result in using police resource effectively and accurately or even to promote public safety in an interactive way.

3 Structuring the data

According to GPS positions, we divided the city of Vancouver in a grid consisting of 13×15 cells. We decided to run a HMM for each of the cells separately.

For a fixed cell, we organized the data in the form of a four dimensional array, containing in the first component the type of crime, in the second component the number of the example and a



(a) type by year

(b) type by month

(c) type by day

(d) type by hour

Figure 1: Histogram data on each time scale with regard to each type of crime.

(days)×(hours) in the last components. Since the days 29-31 do not appear in every month we did not consider them. We have approximately 180 examples for the (days)×(hours) matrices for each crime.

4 Hidden Markov Model

The complexities of the types of crime, the hourly variability and spatial dependence in the data easily escape the untrained eye. We choose to use a Hidden Markov Model (HMM) to try to capture unknown patterns.

An HMM is a directed acyclic graph (DAG) of a specific structure, that assumes that there is a hidden, unobserved Markov chain $\{Z_i : i \in \mathbb{N}_0\}$ according to which the distribution of the actual observations x_i is determined. Figure 2 visualizes the DAG corresponding to an HMM. Our main reference for hidden Markov models was [6].

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and consider a homogeneous hidden Markov chain $Z = \{Z_i\}_{i\in\mathbb{N}}$ and an observation process $X = \{X_i\}_{i\in\mathbb{N}}$, which are collections of the random variables

$$\begin{aligned} Z_i : \quad & \Omega \to E \\ X_i : \quad & \Omega \to \mathbb{N}_{<c}, \end{aligned}$$

where E denotes an abstract state space with $|E| = k$ and $\mathbb{N}_{<c} = \{x \in \mathbb{N}_0 : x < c\}$. The hidden Markov chain is characterized by it's initial distribution $\pi \in \mathbb{R}^k$ and it's transition matrix, which we denote by $A \in \mathbb{R}^{k\times k}$. The variables of the observation process follow a categorical distribution according to the values of the corresponding hidden variables. We denote the emission matrix by $B \in \mathbb{R}^{k\times c}$. As common in machine learning, recalling that we have a homogeneous Markov chain, we use the abbreviation $\mathbb{P}(X_i = x_i) = p(x_i)$ and extend it to multiple events. Note at this point that our model is homogeneous, otherwise we would have to learn the significantly more complex set of parameters $\hat{\Theta} = (\pi, A_1, \dots, A_T, B_1, \dots, B_T)$. This simplifies our model but leaves room for improvement.

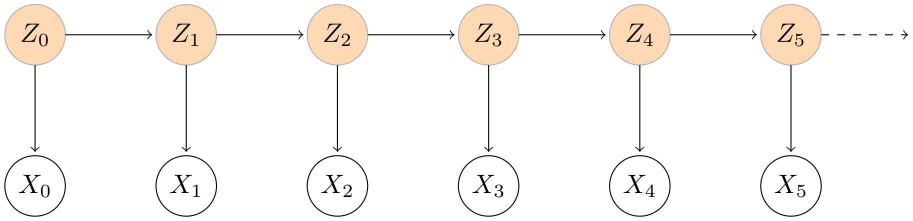


Figure 2: Directed acyclig graph (DAG) corresponding to a hidden Markov model (HMM)

The interference of the DAG is given by

$$p(x_0, \dots, x_T, z_0, \dots, z_T) = p(z_0) \prod_{t=0}^{T} p(x_t|z_t) \prod_{t=1}^{T} p(z_t|z_{t-1}).$$

Given the data, we want our model to *learn* the optimal choice of our model parameters $\Theta = (\pi, A, B)$. However, as half of our random variables are unobserved, we can not perform a standard maximum likelihood estimation. The machine learning technique of expectation maximization (EM) helps out. Instead of maximizing the negative log-likelihood (NLL) we randomly pick initial parameters and maximize the expected NLL with respect to these parameters. As [10] and [12] showed, the algorithm is highly sensitive to it's initialization. We will discuss this later on in more detail.

In a more algorithmic fashion this can be denoted by

$$\begin{aligned} \Theta^{(0)} &= (\pi^{(0)}, A^{(0)}, B^{(0)}) \\ \Theta^{(\ell+1)} &= \operatorname*{argmax}_{\Theta} Q(\Theta^{(\ell)}, \Theta), \end{aligned}$$

where for given realizations $\mathcal{X}, \mathcal{Z}$ of X and Z of length T (sampled according to $\Theta^{(\ell)}$) the function Q is computed by

$$Q(\Theta^{(\ell)}, \Theta) = \mathbb{E}_{\mathcal{Z}|\mathcal{X},\Theta^{(\ell)}} \left[\log(p(\mathcal{X}, \mathcal{Z}|\Theta)\right] = \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{X}, \Theta^{(\ell)}) \log(p(\mathcal{X}, \mathcal{Z}|\Theta)). \tag{1}$$

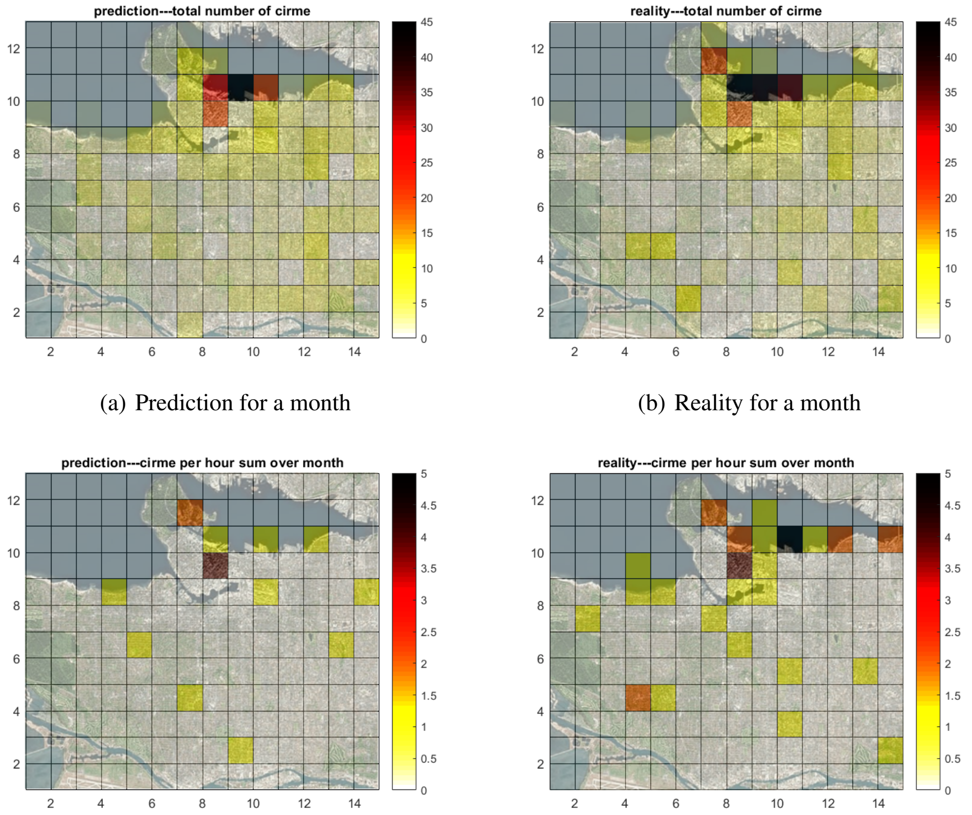
Using the fact that $p(\mathcal{X}, \mathcal{Z}|\Theta) = \pi_{x_0} \prod_{t=1}^{T} a_{z_{t-1},z_t} \prod_{t=0}^{T} b_{z_t,x_t}$ the logarithm on the ride hand side of (1) splits into three terms. Therefore

$$\begin{aligned} Q(\Theta^{(\ell)}, \Theta) = \sum_{\mathcal{Z}} \log(\pi_{x_0}) p(\mathcal{Z}|\mathcal{X}, \Theta^{(\ell)}) + \sum_{\mathcal{Z}} \sum_{t=0}^{T} \log(b_{z_t,x_t}) p(\mathcal{Z}|\mathcal{X}, \Theta^{(\ell)}) \\ + \sum_{\mathcal{Z}} \sum_{t=1}^{T} \log(a_{z_{t-1},z_t}) p(\mathcal{Z}|\mathcal{X}, \Theta^{(\ell)}). \end{aligned} \tag{2}$$

To find the argmax one can now use the structure of the DAG and the resulting conditional independences to compute the terms $p(z_t|\mathcal{X}, \Theta^{(\ell)})$ and optimize each term of (2) with methods of multivariate calculus. This yields the EM-updates

$$\begin{aligned} \pi_i^{(\ell+1)} &= \frac{p(\mathcal{X}, z_0 = e_i|\Theta^{(\ell)})}{p(\mathcal{X}, \Theta^{(\ell)})} \\ a_{ij}^{(\ell+1)} &= \frac{\sum_{t=1}^{T} p(\mathcal{X}, z_{t-1} = e_i, z_t = e_j|\Theta^{(\ell)})}{\sum_{t=1}^{T} p(\mathcal{X}, z_{t-1} = e_i|\Theta^{(\ell)})} \\ b_{ij}^{(\ell+1)} &= \frac{\sum_{t=1}^{T} p(\mathcal{X}, z_t = e_i|\Theta^{(\ell)}) \delta_{x_t,j}}{\sum_{t=1}^{T} p(\mathcal{X}, z_t = e_i|\Theta^{(\ell)})}. \end{aligned}$$

For efficient computation the forward-backward algorithm can be used. This procedure is also known as the Baum-Welch algorithm. An excellent reference for the Baum-Welch algorithm and it's relation to EM provides [14].


prediction---total number of cirme
reality---total number of cirme
prediction---cirme per hour sum over month
reality---cirme per hour sum over month
2
4
6
8
10
12
14
0
5
10
15
20
25
30
35
40
45
0.5
1
1.5
2
2.5
3
3.5
4
4.5

(a) Prediction for a month

(b) Reality for a month

(c) Prediction for a day

(d) Reality for a day

Figure 3: Results of our model for the crime: *mischief*.

5 Results

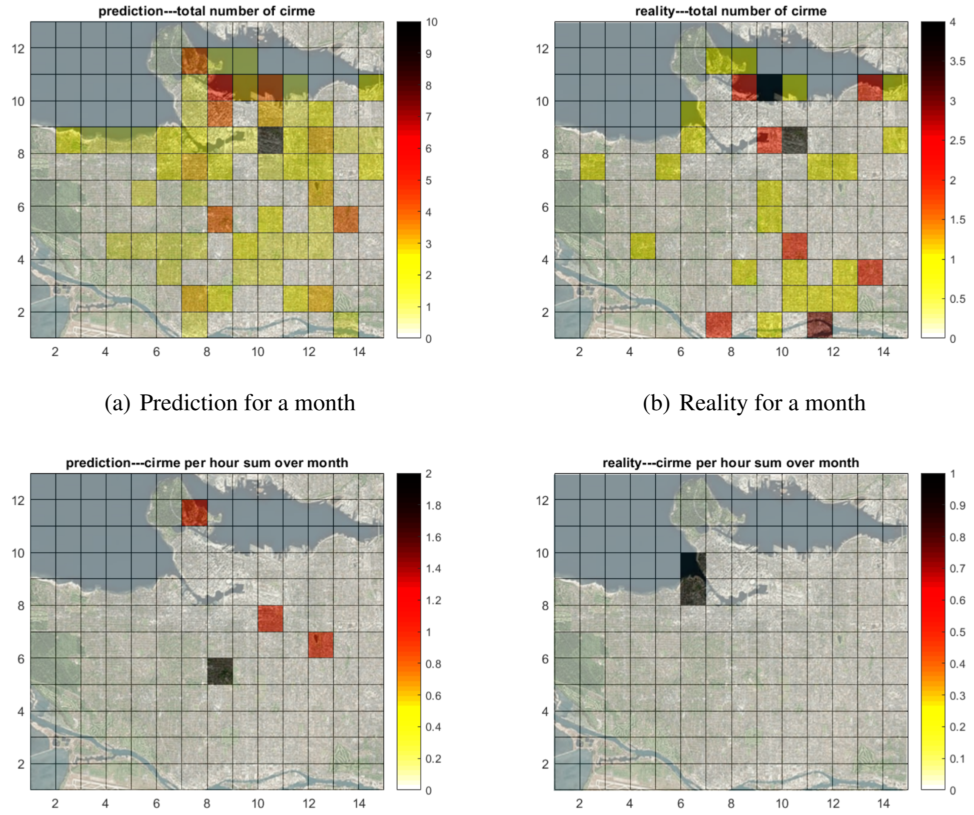
Random samples of our HMM for two specific classes of crimes can be seen in Figure 3 and Figure 4. Our model has learned the distribution of the data very well. Especially when summing up the total amount of crimes committed within the duration of a month, the model seems to perform excellent. However, once less data is given the similarity of prediction and reality is less impressive, which is consistent with the philosophy of the famous law of large numbers.

The significance of our HMM is not to find the area where over time most of the crimes where committed. In fact, a simple averaging model would probably outperform ours when just focusing on the data of the past. What we are interested in is whether there where hidden patterns discovered or not. The answer to this question is found in the matrix B, which tells us how likely it is to see a crime at a fixed location when the hidden chain Z is in a fixed state.

Unfortunately we had to discover that our hidden states don't give an advantage when we try to predict a crime. The rows of matrix B are almost identical. For example, with $k = 10$ we obtained for a specific GPS box the transition matrix

$$B = \begin{bmatrix} .9929 & .9928 & .9929 & .9929 & .9929 & .9928 & .9929 & .9928 & .9928 & .9929 \\ .007099 & .007102 & .007098 & .007099 & .007069 & .007103 & .007082 & .007107 & .007110 & .007074 \end{bmatrix}$$

Even worse, the matrix A that corresponds to the transition probabilities of the hidden Markov chain heavily depends on the initial conditions. Whenever we re-run our algorithm the matrix looks completely different. We already mentioned that [10] and [12] found that this is a problem of HMM in general. Unfortunately we did not have enough time to implement their suggestions for an optimal initialization.


prediction---total number of cirme
reality---total number of cirme
prediction---cirme per hour sum over month
reality---cirme per hour sum over month
2
4
6
8
10
12
14
0
1
2
3
4
5
6
7
8
9
10
0.5
1.5
2.5
3.5
0.2
0.4
0.6
0.8
1.2
1.4
1.6
1.8
0.1
0.3
0.5
0.7
0.9

(a) Prediction for a month

(b) Reality for a month

(c) Prediction for a day

(d) Reality for a day

Figure 4: Results of our model for the crime: *vehicle collision or pedestrian hit*.

6 Future Work & Conclusions

This work contributed to the research in predictive policing by implementing a model that has not yet been explored in this context.

Our model in its current form should not be applied for predictive policing. It might send the police to wrong locations and does not provide an advantage when compared to a simple averaging.

However, we believe that the prediction of our model could be greatly improved by following an inhomogeneous strategy. Due to the limitations of time we where not able to implement such a model.

Furthermore, instead of sampling the x_i from the z_i according to a categorical distribution it might make sense to sample them with respect to a Poisson distribution. This would be reasonable as the decision of a criminal to commit a crime can be assumed to be Binomial distributed and as there are thousands of potential criminals. The natural limit of the Binomial distribution is the Poisson distribution . This would also have the advantage that there could be an arbitrary large number of crimes predicted (of course not in practice due to the limitations of the data types).

Another promising approach might be to use a conditional random field (CRF) or a more general restricted Bolzmann machine (RBM), where the dependencies between the hidden variables are more complex.

Acknowledgments

We would like to thank the Vancouver Police department for providing the data set that was used in our project. Furthermore, we would like to thank professor Mark Schmidt for introducing us to machine learning in a graduate course at the University of British Columbia.

References

[1] Francesco Bartolucci, Fulvia Pennoni, and Brian Francis. "A Latent Markov Model for Detecting Patterns of Criminal Activity". In: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 170.1 (2007), pp. 115–132. ISSN: 09641998, 1467985X. URL: http://www.jstor.org/stable/4623137.

[2] P Jeffrey Brantingham, Matthew Valasik, and George O Mohler. "Does Predictive Policing Lead to Biased Arrests? Results from a Randomized Controlled Trial". In: *Statistics and Public Policy* 5.1 (2018), pp. 1–6.

[3] Rex Chang. *Vancouver Crime Map*. https://www.cs.ubc.ca/~tmm/courses/547-15/projects/rex/report.pdf. Accessed: 2018-04-23.

[4] The Vancouver Police Department. *GeoDash*. http://geodash.vpd.ca. Accessed: 2018-04-26.

[5] The Vancouver Police Department. *Open Data Catalogue*. http://data.vancouver.ca/datacatalogue/crime-data.htm. Accessed: 2018-04-25.

[6] Zoubin Ghahramani. "Hidden Markov Models". In: River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002. Chap. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42. ISBN: 981-02-4564-5.

[7] Hyeon-Woo Kang and Hang-Bong Kang. "Prediction of crime occurrence from multi-modal data using deep learning". In: *PloS one* 12.4 (2017), e0176244.

[8] Alexandra Kim and Amon Ge. "CrimeVis: Visualizing Vancouver Crime". In: *SemanticScholar* (2017).

[9] Lawrence McClendon and Natarajan Meghanathan. "Using machine learning algorithms to analyze crime data". In: *Machine Learning and Applications: An International Journal (MLAIJ)* 2.1 (2015).

[10] Krishna Nathan, Andrew Senior, and Jayashree Subrahmonia. "Initialization of hidden markov models for unconstrained on-line handwriting recognition". In: *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. Vol. 6. IEEE. 1996, pp. 3502–3505.

[11] The Huffington Post. *The Future Of Crime-Fighting Or The Future Of Racial Profiling?: Inside The Effects Of Predictive Policing*. https://www.huffingtonpost.com/entry/predictive-policing-video_us_56f898c9e4b0a372181a42ef. Accessed: 2018-04-26.

[12] Ângela Abreu Rosa de Sá, Alcimar B Soares, and Slawomir J Nasuto. "On the initialization of parameters in hidden Markov models". In: ().

[13] Tong Wang et al. "Learning to detect patterns of crime". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2013, pp. 515–530.

[14] Holger Wunsch. *Der Baum-Welch Algorithmus für Hidden Markov Models, ein Spezialfall des EM-Algorithmus. 2001*. 2014.