CMPT 318

Technical Project Report

Shivanshu Bansal: 301386409

Naman Bharti: 301386412

SM Shajiban Munjoreen: 301323565

Fall 2020

Abstract

With the rapid growth of technology humanity has thrived and made enormous leaps in numerous industries. There was once a time where the thought of going to the moon was just a fantasy and one is this where we carry more computational power inside our pockets. Although, with comparatively less focus on cyber security there lies a great deal of threat to humanity. Through the medium of this paper the authors discuss a statistical modeling technique (Hidden Markov Model) to find patterns in observable data of household electricity consumption and discuss methods such as anomaly detection to get a deeper understanding how one can recognize irregularities with a high degree of precision and accuracy. The applications of such models are limitless, from a worker in a local electricity grid to a real time intrusion detection system. The authors will also discuss some new ways of anomaly detection such as Markov Decision Process which are also widely used in today's time.

Table of Contents

PART 1

1.1 Introduction

In today's world it has become extensively hard to monitor, ethically, the behavior of not only the people in our community but the community as a whole. With the rapid growth of technology and minimal focus on securities and exploitation, we have given tremendous amount of power in the hands of almost the entire habitation of the planet Earth. Today anyone anywhere on the face of this planet has the capability to go dark and undetectable on the internet and carry out all sorts of actions. In many instances or professions such as journalism and politics this anonymity is required to maintain safety and peace whereas in some, it is a necessity to being about a restoration of peace. But just like anything, if there is a positive use case of a tool there is also the possibility of a negative use case. One negative side to anonymity is cybercrime and terrorism where anonymity is the enemy of stability and peace. Today more acts of crime are being carried out over the internet away from the peering eyes of authorities rather than on the streets. These acts don't require big corporations or offices. They can be carried out from inside a house which even many trained eyes cannot depict at first glance.

This raises some serious concerns about the safety and security of everyone. If we look at the job market sector for cyber security, we will find that at any given time the ratio of job openings to total current workforce is about 1:2. In laymen terms this means that the cyber security industry is heavily understaffed.

Now, suppose you are a worker at the local electricity grid and are tasked to find suspicious activity in the community. To the naked eye everything might seem to be peaceful and out and about in the community but if you start monitoring the household electricity consumption of the neighborhood, you'll come to find some anomalies in the data. These anomalies can be anything from just a resident that likes to party every day of the week to a cybercriminal warehouse. Throughout this paper we will discuss about how tools such as Hidden Markov Models (HMM) can be used to help us find these sorts of anomalies and in turn help us prevent these disgraceful acts of crime. An HMM is statistical Markov model in which the system being modeled is assumed to be with unobservable number of states and assumes that there is a process that depends on these states. The mere goal or ideology is to learn and interpret by the act of observing.

1.2 Classification of Data

We started out by separated our data into two different sets consisting of weekdays and weekends as people consume electricity in different patterns on weekdays when compared to weekends and vice versa. After that, we further divided our four-year data into two parts: three-year data and one-year data so that we can train our model on the first three years and then test the model on last year. Using three years to train our model was a good choice as it gives us sufficient amount of data to create a model that provides a good depiction of the electricity consumption patterns in households. The rest of the data (last one year) would help us in choosing the best model meaning a model that is neither underfitted nor overfitted and thus provides a good interpretation of the patterns followed by the data.

Usually, on weekdays people reach their homes by 06:00 P.M. and start using various appliances at home to make food or watch TV. This results in increasing of electricity consumption by 06:00 P.M. By 10:00 P.M. on weekdays, people start to wind down and go to bed as they have to go work or school the next day and are too tired from all the work and interesting activities being carried out during the day. This results in the decrease of electricity consumption by 10:00 P.M. On weekends, people typically spend their time at home focusing more on entertainment and leisurely activities which leads to an increase in electricity consumption. Although by 10:00 P.M. on weekends, people start to go to bed as either they have to maintain their weekly schedule or go to work or school the next day.

Therefore, we chose the time window from 06:00 P.M. to 10:00 P.M. for both, weekdays and weekends. By doing this we are able to get a good variation in observations of data which can help us in creating a well fitted model.

1.3 Selection of Response Variables using PCA

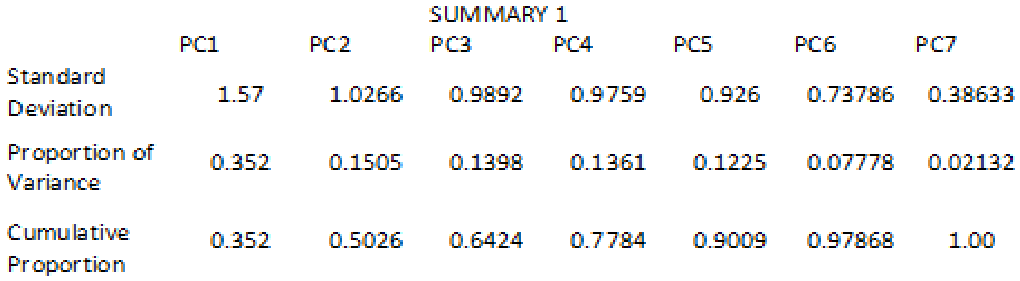
1.3.1 What is PCA?

Principal Component Analysis is a technique that is extremely useful in analyzing datasets that contain a multitude of variables. It simplifies the dataset by turning it into smaller number of variables called principal components. This method gives us a number for each response variable which tells us how much variance there is for each response variable which helps us to visualize the spread of response variables in our data.

1.3.2 Response Variable Selection

Selection for Weekdays:

After using prcomp() function in R for calculating the PCA of response variables of weekdays, we get the following tables.



SUMMARY 1

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard Deviation	1.57	1.0266	0.9892	0.9759	0.926	0.73786	0.38633
Proportion of Variance	0.352	0.1505	0.1398	0.1361	0.1225	0.07778	0.02132
Cumulative Proportion	0.352	0.5026	0.6424	0.7784	0.9009	0.97868	1.00

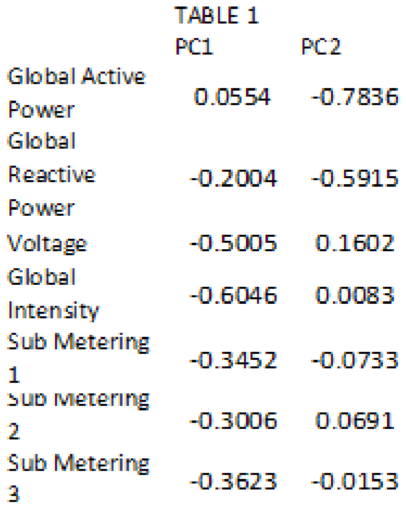
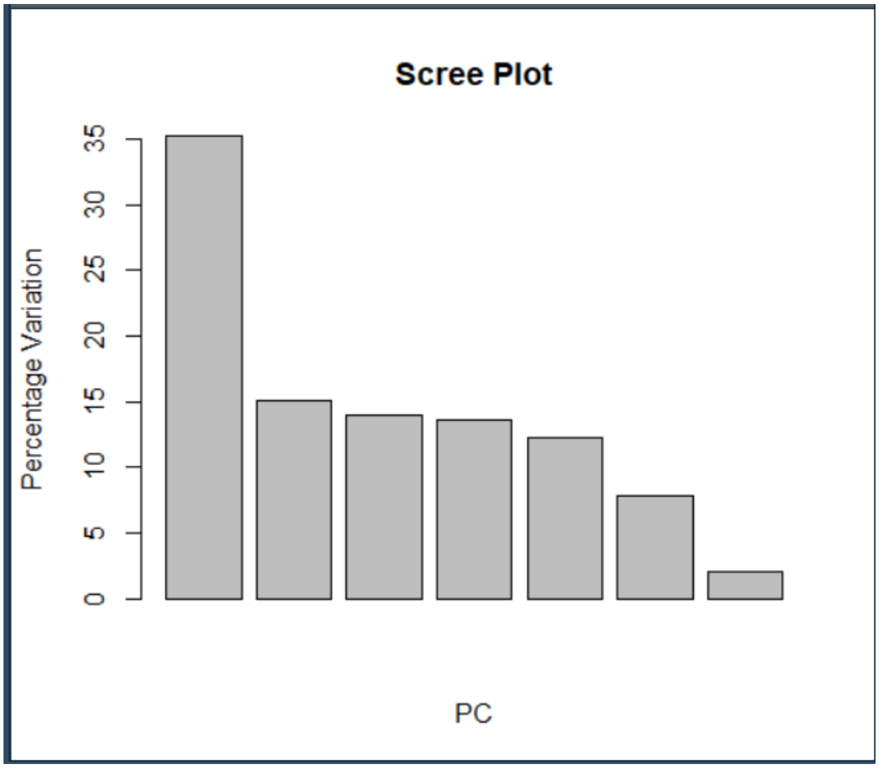


TABLE 1

	PC1	PC2
Global Active Power	0.0554	-0.7836
Global Reactive Power	-0.2004	-0.5915
Voltage	-0.5005	0.1602
Global Intensity	-0.6046	0.0083
Sub Metering 1	-0.3452	-0.0733
Sub Metering 2	-0.3006	0.0691
Sub Metering 3	-0.3623	-0.0153

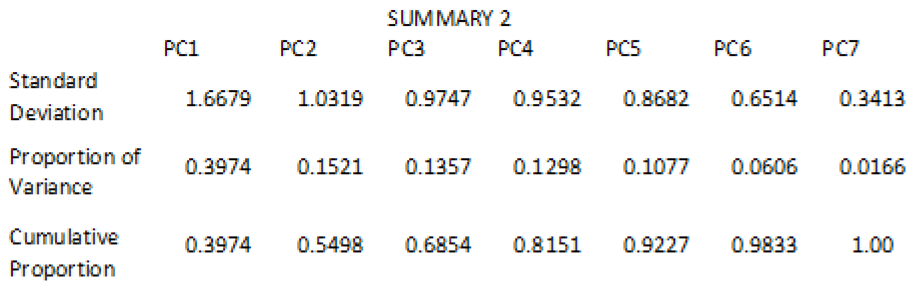
The following graph shows us that PC1 has the largest proportion of variance and hence it is most likely that the response variables to be chosen will be influenced from PC1.



After doing several calculations, we get 0.21 for Global Intensity from PC1, 0.18 for Voltage from PC1 and 0.11 for Global Intensity from PC2. As it can be seen that Global Intensity and Voltage have large numbers when compared to any other response variable from any other PCs, and therefore we will take Global Intensity for Univariate modelling and Global Intensity and Voltage for Multivariate modelling for weekdays.

Selection for Weekends:

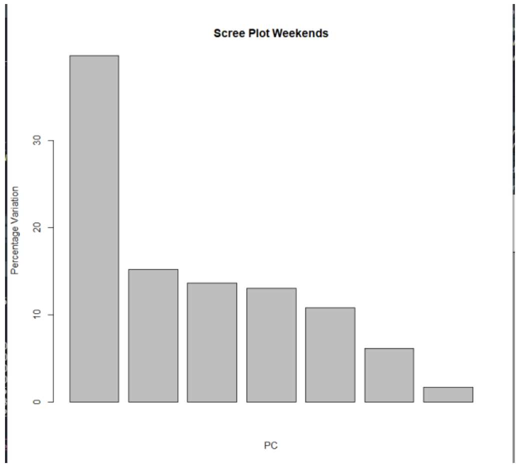
After using prcomp() function in R for calculating the PCA of response variables of weekends, we get this table.



SUMMARY 2

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard Deviation	1.6679	1.0319	0.9747	0.9532	0.8682	0.6514	0.3413
Proportion of Variance	0.3974	0.1521	0.1357	0.1298	0.1077	0.0606	0.0166
Cumulative Proportion	0.3974	0.5498	0.6854	0.8151	0.9227	0.9833	1.00

The following graph show that PC1 has the largest proportion of variance therefore we can come to a conclusion that any probability of variables in PC1 (Table1) when multiplied by the proportion of variance of PC1 will result in a number that is greater than a result derived by the same procedure for any other principal component and their respective variables. Hence, it is most likely that the response variables to be chosen will be influenced from PC1.



We get similar results from weekdays in weekends for various response variables for PC1 and PC2. Hence, we will take Global Intensity for Univariate modelling and Global Intensity and Voltage for Multivariate modelling for weekdays and weekends. This may also help us to compare models in the future as we are taking same response variables for both weekdays and weekends.

1.4 Training and Testing of Data

This section will cover all the details and procedures that were followed in the development of both Univariate (1.4.1) and Multivariate (1.4.2) Hidden Markov Models.

The model was created using the depmix framework which is used for specifying and fitting Hidden Markov Models. Next the fit() function was used during training of each of the models (both for Univariate and Multivariate) to impose the constraints and get the Bayesian Information Model (BIC) and the Log Likelihood of the model.

The BIC is defined to be an estimate the posterior probability of a model being true. Therefore, a lower BIC value indicates that the model is to be considered as more likely to be a true model.

Log Likelihood is a function that is used to measure the goodness of a fit of a statistical model to a sample of the data. Therefore, a higher Log Likelihood value suggests that our model is essentially a good fit for our sample data and henceforth is a good model.

We also have to normalize the log likelihoods of both, the trained models and the test models. This is done to ensure that we are making a comparison of apples vs apples and not apples vs oranges. It is clear that the training dataset tracks the household electricity consumption of over three years; whereas our testing data tracks only one year, which is significantly short. To ensure a balanced comparison we normalize both log likelihoods. To determine the normalized value we divide the log likelihood of the model by the number of observations.

1.4.1 Univariate Modeling

Univariate time analyses are extensively used in researching the quality of life. It is also often defined as an analysis of a single variable in order to summarize it and describe its pattern.

For this modeling Global Intensity variable was chosen, as its value in PCA1 is the highest. Further and more elaborate explanation can be found in section 1.3 Selection of Response Variables using PCA of this report.

1.4.1.1 Training

For the training of our univariate weekdays model, we chose the number of states to be 10, 13 and 17. We chose these numbers because they provide a good variation between the range of 10 to 20, where it is most likely to get to good fit for our model.

	Training Weekday	Univariate	
# of States:	10	13	17
LogLik	-168,943.5	-149,229.6	-139,181.6
BIC	339,286.4	300,740.4	282,149.7
AIC	338,125.1	298,847.2	279,007.2

As it can be seen from the above table, we get the best balance between BIC and log likelihood from the model that corresponds to 17 number of states. This means that the model from 17 number of states can potentially be a good fit model.

For the training of our univariate weekends model, we chose the number of states to be same as weekdays as these number of states had given us a good variation in BIC and log likelihood in weekdays training of models.

	Training	Weekend	Univariate
# of States	10	13	17
Log Lik	-75,297.88	-69,391.88	-40,525.6
BIC	151,686.5	140,888	84543.82
AIC	150,633.8	139171.8	81695.21

As it can be seen from the above table, we again get the best balance from the model that corresponds to 17 number of states. This means that the model can be a good fit model.

Now, we will confirm the fit of these models by testing the models against a data of one year that had been previously separated from the large dataset.

1.4.1.2 Testing

For weekdays, after normalizing the log likelihoods of both the train and test datasets, we get a difference of 0.005908501 which confirms that the model that we got from 17 number of states is a good fit model.

Similarly, for weekends, after normalizing the log likelihoods of both the train and test datasets, we get a difference of 0.000466885 which ensures that the model that we got from 17 number of states is a good fit model.

These results imply that we can use these models in the future to study different datasets for finding various outcomes such as estimating the number of anomalies in datasets.

1.4.2 Multivariate Modeling

A multivariate analysis is utilized to study datasets that are much more complex compared to what the methods of a univariate model can handle. It is pretty much impossible for someone to work with such datasets with hand. Multivariate statistics is also a subdivision of statistics wherein a simultaneous observation and analysis of more than one outcome variable is carried out.

For this model, Global Intensity and Voltage were chosen as the response variables due to their high outcomes in the principal component's analysis. Further and more elaborate explanation can be found in section 1.3 Selection of Response Variables using PCA of this report.

1.4.2.1 Training

For the training of our multivariate weekdays model, we chose the number of states to be 10, 13 and 17. We chose these numbers because they provide a good variation between the range of 10 to 20, where it is likely that we can see a good fit of model.

	Training Weekday	Multivariate	
# of States	10	13	17
Log Lik	-452,818.7	-434,230.8	-415,968.9
BIC	907,271.6	871,047.9	836,123.1
AIC	905,915.4	868,901.6	832,649.9

As it can be seen from the above table, we get the best balance between BIC and log likelihood from the model that corresponds to 17 number of states. This means that the model from 17 number of states can potentially be a good fit model.

For the training of our multivariate weekends model, we chose the number of states to be 10, 13 and 17 in the beginning because we believed that they can provide a good variation between the range of 10 to 20, where it is likely that we can see a good fit of model. After training for 17 number of states, we trained our model for 19 number of states because we assumed that we can get a better balance between BIC and log likelihood for 19 number of states.

	Training	Weekend	Multivariate	
	10	13	17	19
Log Lik	-184,138	-177,309.3	-167,568.8	-263,252.4
BIC	369,781.6	357,001.8	338,993.9	329,025.8
AIC	368,553.9	355,058.7	335,849.6	325,174.8

As it can be seen from the above table, we get the best balance between BIC and log likelihood from the model that corresponds to 17 number of states and even the 19 number of states did not give the best balance and seems to be an overfit. This means that the model from 17 number of states can potentially be a good fit model.

Now, we will confirm the fit of these models by testing the models against a data of one year that had been previously separated from the large dataset.

1.4.2.2 Testing

For weekdays, after normalizing the log likelihoods of both the train and test datasets, we get a difference of 0.3178618 which confirms that the model that we got from 17 number of states is a good fit model.

Similarly, for weekends, after normalizing the log likelihoods of both the train and test datasets, we get a difference of 0.3465053 which ensures that the model that we got from 17 number of states is a good fit model.

Now, we will use the models that we got from multivariate weekdays and weekends modelling to model three datasets so that we can estimate the number of anomalies that are injected in them.

1.5 Anomaly Detection

Anomaly detection is about finding patterns in datasets that do not conform to the normal behavior in datasets. These patterns are called anomalies. Anomalies are also usually referred to as outliers, exceptions or contaminants. Anomaly

Detection can be used for various purposes such as detecting fraud for credit cards, intrusion detection for cybersecurity, military surveillance or counterterrorism.

It is often challenging to find anomalies in datasets due to several reasons. One reason is that the boundary between normal and anomalous behavior is often not precise which makes it hard to detect them. Another reason is that in several situations the normal behavior keeps evolving and the present notion of normal behavior might not represent the future.

In this project, we were given 3 datasets and the task to estimate anomalies in them. We used the method of Log Likelihood to detect anomalies in these datasets. The idea is that we will calculate the normalized log likelihoods of these 3 datasets. The lesser the log likelihood will mean that dataset will have fewer anomalies when compared to other datasets. Now, we will look at the observations from these 3 datasets and estimate the degree of anomalies in them.

1.5.1 Observations from 3 Datasets

Observations for Weekdays:

	Weekdays	17 states	
	Dataset 1	Dataset 2	Dataset 3
Normalized Log Likelihood	-3.442807	-3.512262	-3.442807

Observations for Weekends:

	Weekend	17 states	
	Dataset 1	Dataset 2	Dataset 3
Normalized Log Likelihood	-3.279567	-3.339795	-3.279567

1.5.2 Conclusion from Anomaly Detection

For both weekdays and weekends, it can be clearly seen that the normalized log likelihoods for dataset 1 and dataset 3 are same and the log likelihood of dataset

2 is larger than both of them. This implies that the degree of anomalies present in dataset 2 is larger than the degree of anomalies present in dataset 1 and dataset 3. It also implies that the degree of anomalies present in dataset 1 and dataset 3 are same. Therefore, it can be said that there are more anomalies in dataset 2 than dataset 1 and dataset 3. Also, dataset 1 and dataset 3 will have approximately same number of anomalies in them.

In the next section, we will learn some more ways for anomaly detection.

Part 2

2.1 Introduction

As AI becomes more of an everyday thing now, those who can are taking full advantage of it. Organizations are using it to further their research as well as protect their research and maintain security. At the same time criminals are also trying new methods access and steal unauthorized data. The world is facing increasing amounts of AI powered cyber-attacks everyday [1] therefore organizations are implementing different ways to maintain security. To fight various cyber-attacks and computer viruses, many computer security techniques have been studied including firewalls and intrusion detection. Among these many methods, intrusion detection is the most viable to defend against complex intrusive behaviors [2]. This technique consists of developing behavior models and patterns to detect intrusion. So now, of the core problem is to come up with good behavior model and patterns that will go through large amounts of data and find differences between normal behavior and unexpected behavior accurately and efficiently.

2.2 Early models

One of the first intrusion detection models were drawn by D.E. Denning [3]. His model was constructed based on the hypothesis that security violations can be spotted by observing a system's audit records for irregular patterns of system usage. This model of his comprised of profiles for demonstrating the behavior of subjects with respect to objects in terms of metrics and statistical models, and rules for acquiring knowledge about this behavior from audit records and for detecting anomalous behavior [3]. Since then, many hours of research have been based off of this hypothesis. Researchers have tried to come up with the most effective form of intrusion detection models.

2.3 Current Approach

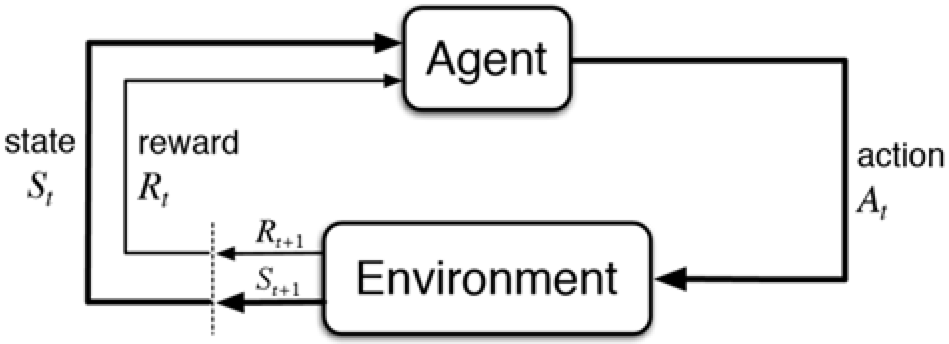
Till date, there are two general approaches to constructing intrusion detection models- misuse detection and anomaly detection. Misuse detection extracts behavior pattern based on the audit data of identified attacks so that novel attacks may fail to be uncovered by misuse detection systems. Anomaly detection established normal behavior models to recognize suspicious attack activities. Although new attacks can be spotted by anomaly detection systems, increased rates of false alarms regularly occur. The goal is to reduce the number of false alarms as much as possible.

2.4 Anomaly Detection Systems

This paper will be focusing more on anomaly detection systems. Recently the application of machine learning and data mining in anomaly detection systems has been taken into the attention of many researchers [4]. By using machine learning algorithms, adaptive anomaly detection models can automatically be constructed based on labeled or unlabeled audit data [4]. For an anomaly detection system to work in this manner, a machine learning model is trained with long lists of normal expected data. With time the model gets better at identifying anomalies. As security measures get stronger and more sophisticated, criminals must do the same with their attacks to keep up. Their methods of attack keep changing too. For a machine learning model to work in this scenario, the training must be continuous. This, in theory can achieved by an online learning system. With time, more data over the internet will be fed into the machine learning model and it will keep getting better at recognizing anomalies. This is also work better because data over the web will be much more recent therefore more relevant. In practice though, it is not as simple as it sounds.

2.5 Markov Decision Process

The Markov Decision process involves evaluative feedback and an associative aspect—choosing different actions in various situations. They are a classic form of sequential decision making, where actions impact not just instant rewards, but also succeeding situations, or states, and through those future rewards. Thus, Markov Decision Processes involve delayed reward and the need to interchange immediate and delayed reward. [5]



[5]

The figure above shows the core idea behind reinforced learning. There exists an environment which represents the outside world to the agent. There also exists an agent that takes actions and receives observations from the environment that involves a reward for its action and information of his new state. That reward notifies the agent if the action taken was good or bad, and the observation tells him his new state in the current environment.

The agent tries to figure out the optimal course of actions to take or the best way to behave in the environment in order to carry out its task in the most desirable way possible. [6]

More specifically, the agent and environment communicate at each of a sequence of distinct time steps, t = 0, 1, 2, 3, ... At each time step t, the agent receives some depiction of the environment's state, $S_t \in S$, and on that basis selects an action, $A_t \in A(s)$. One time step later, as a consequence of its action, the agent receives a numerical reward, $R_{t+1} \in R \subset R$, and finds itself in a new state, S_t+1. The MDP and agent together thus create a sequence or trajectory that begins like this: S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ... [5]

The agent looks to maximize the function of future rewards known as the function. It means different things depending on the type of task and whether you want to discount delayed reward. The undiscounted formulation is fitting for discontinuous tasks, in which the agent–environment interaction is divided naturally into episodes; the discounted formulation is appropriate for continuing tasks, in which the interaction does not naturally divided into episodes but continues without limit. Researchers try to define the returns for both kinds of tasks so that one set of equations can apply to both the episodic and continuing cases. [5]

2.6 Examples

An example is a self-driving school bus that has the job of picking up children and taking the, all to a school. It has an address book containing location of all the students and the school. It has sensors for detecting streets, dividers, other vehicles and people. It runs on a rechargeable battery. The bus's control system has mechanisms for interpreting sensory information, for traversing, and for controlling the wheels. High-level decisions about how to drive are made by a reinforcement learning agent based on the current charge level of the battery. This agent has to decide whether the robot should

(1) actively drive for a certain period of time,

(2) remain stationary and wait for someone to board the bus, or

(3) head back to its home base to recharge its battery.

This decision has to be made either periodically or whenever certain events occur, such as pulling up to a student's house or the school. The agent thus has three actions, and the state is primarily determined by the state of the battery. The rewards might be zero most of the time, but then become positive when a student gets on or off the bus, or large and negative if the battery runs all the way down. In this example, the reinforcement learning agent is not the entire bus. The states it monitors describe conditions within the bus itself, not conditions of the bus's external environment. The agent's environment therefore includes the rest of the bus, which might contain other complex decision-making systems, as well as the bus's external environment.

Another example is a mobile robot that has the job of collecting empty soda cans in a mall. It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin; it runs on a rechargeable battery. The robot's control system has mechanisms for interpreting sensory information, for traversing, and for controlling the arm and gripper. High-level decisions about how to search for cans are made by a reinforcement learning agent based on the current charge level of the battery. This agent has to decide whether the robot should

(1) actively search for a can for a certain period of time,

(2) remain stationary and wait for someone to bring it a can, or

(3) head back to its home base to recharge its battery.

This decision has to be made either periodically or whenever certain events occur, such as finding an empty can. The agent thus has three actions, and the state is primarily determined by the state of the battery. The rewards might be zero most of the time, but then become positive when the robot secures an empty can, or large and negative if the battery runs all the way down. In this example, the reinforcement learning agent is not the entire robot. The states it monitors describe conditions within the robot itself, not conditions of the robot’s external environment. The agent’s environment therefore includes the rest of the robot, which might contain other complex decision-making systems, as well as the robot’s external environment. [5]

2.7 Conclusion

A reinforcement learning problem can be modelled in various different ways depending on assumptions about the level of knowledge primarily available to the agent. In problems of complete knowledge, the agent has a complete and accurate model of the environment’s dynamics. If the environment is a Markov Decision Process, then such a model comprises of the complete four-argument dynamics function p. In problems of incomplete knowledge, a complete and perfect model of the environment is unavailable.

Even if the agent has a complete and accurate environment model, the agent is usually unable to perform sufficient calculation per time step to fully use it. The memory available is also an important limitation. Memory may be required to build up accurate approximations of value functions, policies, and models. In most cases of practical interests, approximations must be made because there are too many states than entries can be made in a table.

3 Bibliography

[1] S. Bocetta, "Has an AI Cyber Attack Happened Yet?," 10 March 2020. [Online]. Available: https://www.infoq.com/articles/ai-cyber-attacks/. [Accessed 25 November 2020].

[2] X. Xu and T. Xie, "A Reinforcement Learning Approach for Host-Based Intrusion Detection Using Sequences of System Calls," *Advances in Intelligent Computing,* vol. 3644, pp. 995-1003, 2005.

[3] D. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering ,* Vols. SE-13, no. 2, pp. 222-232, 1987.

[4] W. Lee, S. Stolfo and K. Mok, "A data mining framework for building intrusion detection models," *Proceedings of the 1999 IEEE Symposium on Security and Privacy,* 1999.

[5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge: The MIT Press, 2017.

[6] M. Ashraf, "Reinforcement Learning Demystified: A Gentle Introduction," 7 April 2018. [Online]. Available: https://towardsdatascience.com/reinforcement-learning-demystified-36c39c11ec14. [Accessed 2 December 2020].