Simon Fraser University

CMPT-318

Term Project Report

Group 14

Kwok Yee Cheung (301367833)

Junchen Li (301385486)

Nov 29, 2020

Abstract: The focus of this report is to analyze a set of “real-world” power consumption data and train univariate and multivariate Hidden Markov Models (HMM) via Principal Component Analysis (PCA). It uses models to detect anomalies dataset and picks the suitable model which can cover more abnormal data from the dataset.

Table of Contents

Introduction

Project Description:

In the era of automation, discovering and quickly dealing with cyber threats and security vulnerabilities has become the focus of attention. Some uncontrollable reasons have caused the data to show an abnormal state. In an environment with limited resources, the abnormal data have utilization values, which can better deal with network security issues.

Based on the characteristics of given electricity consumption data, find and construct HMM models for anomaly detection. By using the PCA analysis method, both univariate and multivariate HMM models were constructed on the normal dataset without anomalies. Using the multivariate HMM, compute the log-likelihood and compare log-likelihood for three different sets of abnormal data.

For the respective observation sequences associated with the same time window in each of three test datasets with injected anomalies, it chooses a suitable model that shows normal system operation behaviours under different scenarios of anomalies.

Evaluation:

Use the results of log-likelihood and Bayesian information criterion of a sequence of observation to measure the training model accuracy. In addition, use the percentage of the total variation in the dataset to determine the principal components.

Data Variables Description：

This experiment uses a standard electricity consumption summary database. The time period is from 2006 to 2009. There are three data sets with abnormal electricity consumption. The period is from the 1st to the 8th of December 2009. The four databases all record data of nine variables with one minute as a time node. The nine variables in each database are:

1. Date: Date information recorded in the form of days/months/years.

2. Time: Time information recorded in the form of hours/minutes/seconds.

3. Global active power: The average actual power consumption of the resistance circuit per minute in the home world. The actual work done in the load is usually measured in kilowatts.

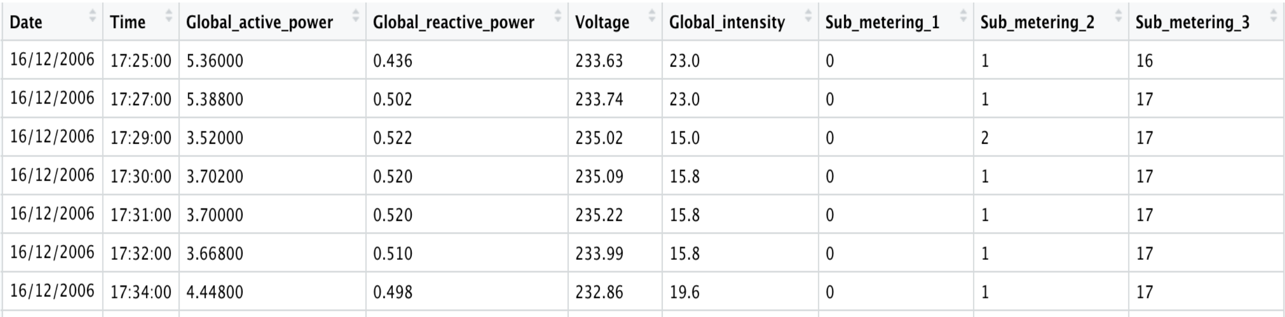
4. Global reactive power: The sum of the average inductive and capacitive power consumed by the family every minute in the world. The energy exchange between the power supply and the reactive load is measured in kilowatts.

5. Voltage: A pressure from a circuit power supply can push charged electrons so that they can do work.

6. Global intensity: Record the global average current intensity per minute of the household in amperes.

7-8-9 Three energy power record tables, which record three different areas where each household's power consumption is concentrated in units of active power.

The following is a small snippet of the electricity consumption data



Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
16/12/2006	17:25:00	5.36000	0.436	233.63	23.0	0	1	16
16/12/2006	17:27:00	5.38800	0.502	233.74	23.0	0	1	17
16/12/2006	17:29:00	3.52000	0.522	235.02	15.0	0	2	17
16/12/2006	17:30:00	3.70200	0.520	235.09	15.8	0	1	17
16/12/2006	17:31:00	3.70000	0.520	235.22	15.8	0	1	17
16/12/2006	17:32:00	3.66800	0.510	233.99	15.8	0	1	17
16/12/2006	17:34:00	4.44800	0.498	232.86	19.6	0	1	17

Analyze tool:

PCA:

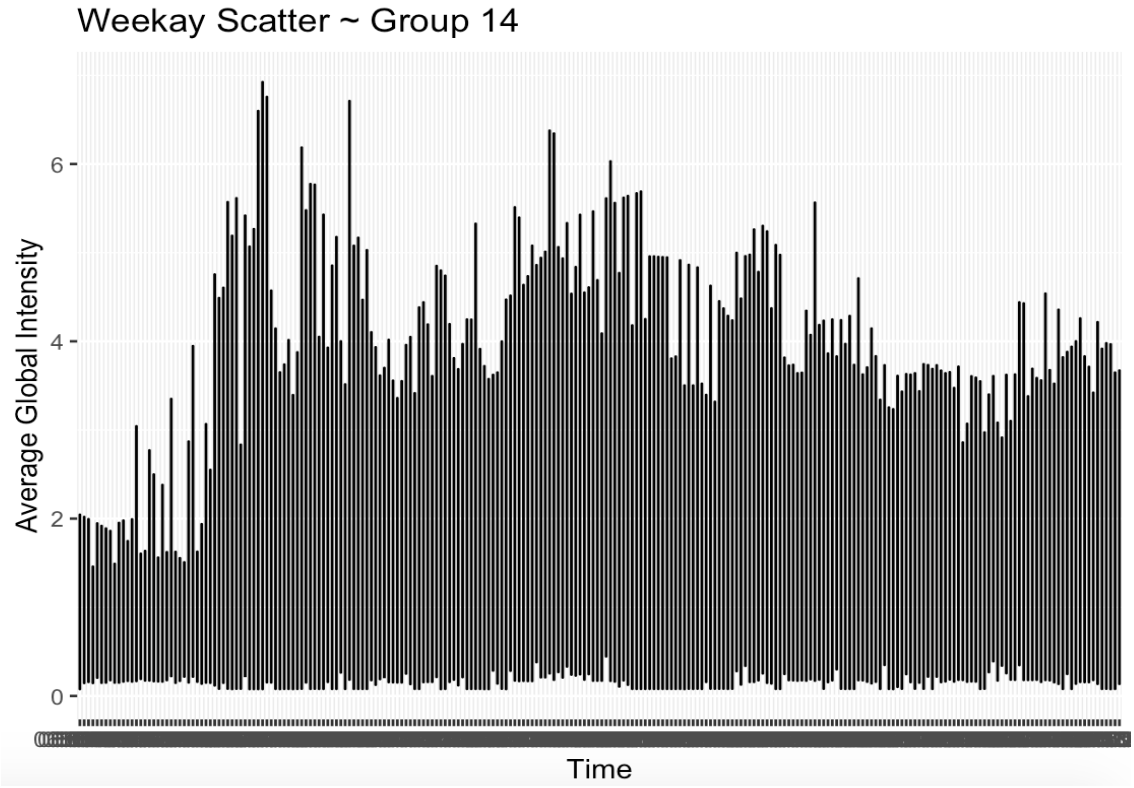
Network security analysis often analyzes a large number of data in different dimensions. The characteristics of high-latitude data are dependent of each other, and they will interfere with and affect the performance of the algorithm. The most serious point is that the spatial distribution of high-latitude data samples will become very sparse, which will lead to overfitting.

In this data analysis, principal component analysis (PCA) will be used for feature engineering. The main purpose of PCA is to find and replace the original data with the most important aspect of the data. For example, an n-dimensional data set with m data. The analyst hopes to reduce the dimensions of m data from n to n', and expects the m data of n dimensions can also represent the characteristics of the original data set. PCA is a means of dimensionality reduction and a balance between latitude reduction and data interpretation capabilities. The specific principle is to find a new set of lower-dimensional bases in the sample space, and project the original data on this new set of bases (express each sample as a linear combination of this set of bases). Reducing the dimensionality will cause a loss of data and using PCA can minimize the loss.

Design

Time window selection:

For training univariate and multivariate Hidden Markov Models on normal electricity consumption data, the data time window chosen is from 6 am to 9 am every Thursday. The plot function is used to plot out the selected data. It gives an overview of how the average global intensity data looks like. The image is displayed below:

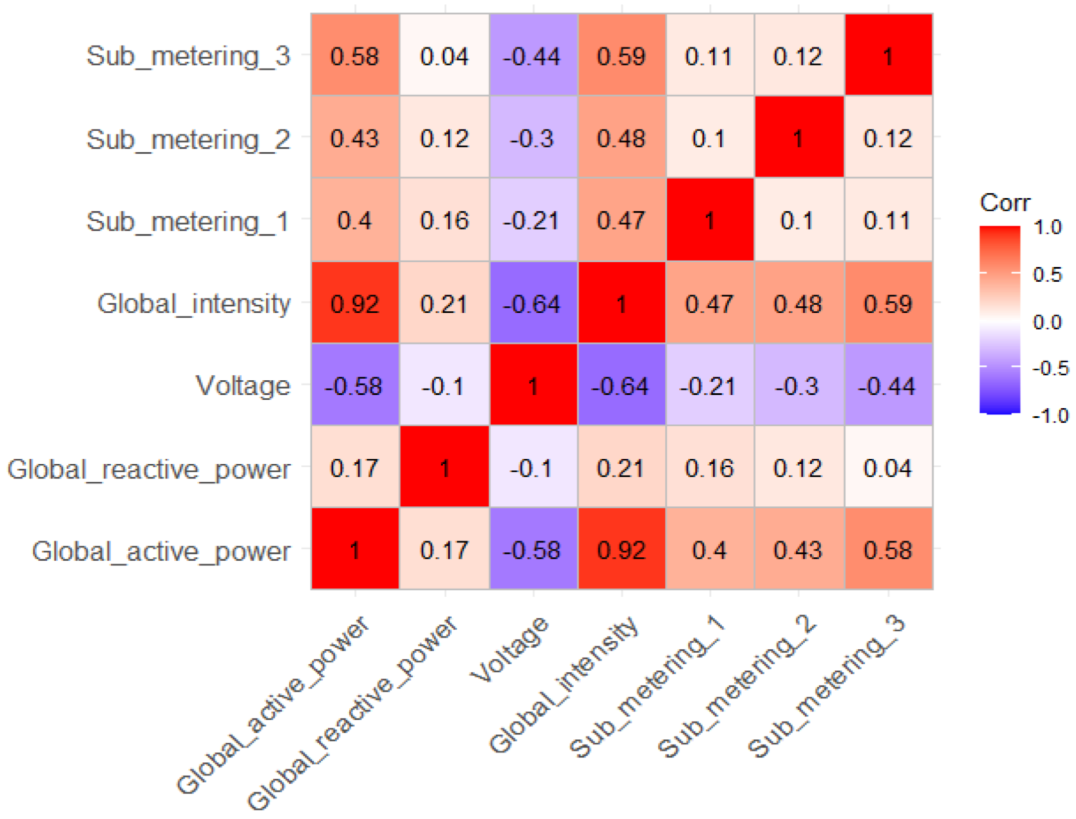


Variables selection:

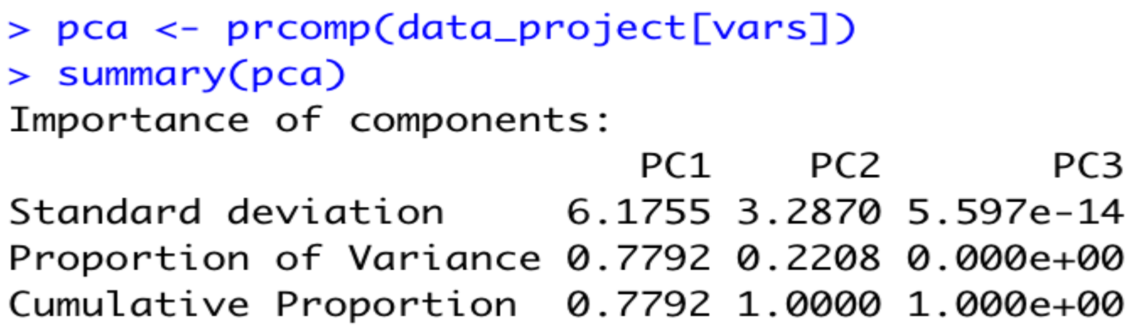
To know which of the 9 data variables has the greatest impact on Global_active_power, calculate the correlation among those 9 and get all the results in the range of [-1, +1]. Find the number that corresponds to the target variable closest to the positive one. As shown in the figure below, the closer the colour is to the red, the stronger the correlation. Although the value of Global_intensity is closest to +1, it cannot be selected as a research variable for the following reasons:

1) If variable Global_intensity is selected, it will grab a correlation with other variables. In other words, even if all the variables are selected, the Global_intensity variable is enough to be the principal component in the PCA, which results in PCA multivariate HMM model not constructed.

2) To be able to compare with the univariate HMM model and multivariate HMM model, the selection must be multivariate and have a connection between the selected variables.



This report uses the Principle Components Analysis function(procomp()) to analyze the components after choosing the PCA variables. The image below is the information on using the summary function on PCA.

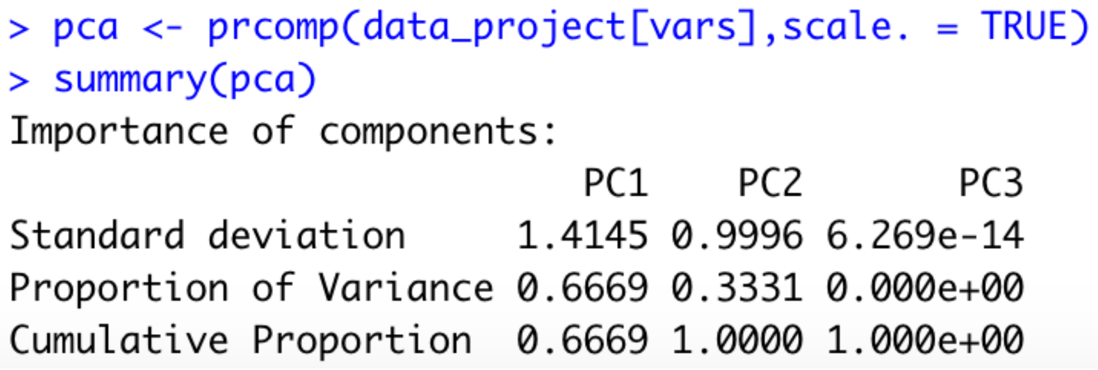


```
> pca <- prcomp(data_project[vars])
> summary(pca)
Importance of components:
                          PC1    PC2       PC3
Standard deviation     6.1755 3.2870 5.597e-14
Proportion of Variance 0.7792 0.2208 0.000e+00
Cumulative Proportion  0.7792 1.0000 1.000e+00
```

Problem Analysis:

1) To reasonably determine the number of factors for the PCA multivariate model. The factor is the principal component because in the principal component extraction, the rotation operation is performed through the factor load matrix and convergence is achieved. Preliminary analysis using the latter shows that the cumulative proportion of the first principal component is 66.69%, and the proportion of the second principal component is 100%. Therefore, for this analysis, the first two principal components can cover the original three variables, making the third principal component completely unnecessary. After the PCA process, the dimension of the data has been successfully changed from 12480x13 to 12480x2.

2) In the subsequent calculations, it was found that the variance of PC1 and PC2 was too large, especially since the variance of PC1 was as high as 6. When searching for the PCA materials, it was found that the value of the function parameter "scale" was ignored. The solution is to set the parameter "scale = TRUE" when using the prcomp function. The reason for setting the parameter "scale = TRUE" is to standardize the input data. In order to decrease both the mean and variance to be 1 before PCA analysis. This action is also known as standardization. Therefore, "prcomp" function is used to analyze data again after using the scale attribute. The result is demonstrated below:



```
> pca <- prcomp(data_project[vars],scale. = TRUE)
> summary(pca)
Importance of components:
                          PC1    PC2       PC3
Standard deviation     1.4145 0.9996 6.269e-14
Proportion of Variance 0.6669 0.3331 0.000e+00
Cumulative Proportion  0.6669 1.0000 1.000e+00
```

As can be seen from the figure aboe, the cumulative proportion of the first principal component accounts for 66.69%, and the cumulative proportion of the second principal component has reached 100%. This is consistent with the summary of the previous analysis. Through the two comparisons, it can be found that the proportion of the first principal component after the scale is reduced because after the standardization, the influence of the dimension on the extraction of the principal component is removed.

3) After performing the feature extraction projects, this can achieve the establishment of a relevant HMM model and compare the model results. To prevent the model from overfitting, it is a good choice to divide all data into the training set and test set. The training set is used for model training, including parameter adjustments (such as finding the optimal parameters on the training set). The test set is used for data testing and inspection to see the final effect of the model. Through the advice given by the Machine Learning Crash Course: "Data is divided into training sets and test sets in an 80-20 split manner. After training, the model achieves 99% accuracy in both the training set and the test set." Set the segmentation coefficient to 0.8, set the data less than or equal to 0.8 as the training set, and set the data greater than 0.8 as the test set. This suggestion solves the data proportional distribution problem.

4）There is a problem with the use of data classification functions for training and testing. If the sample function is used for random sampling, although the expected effect can be achieved, the seed value needs to be set if the sample appears repeatedly. Different computers will have differences, so use the runif function as a solution. Runif is also r uniform, and its function is to generate a uniform distribution and store it as a column in the data. It is a 0-1 uniformly distributed random value, which is stored to ensure that the data can be reproduced.

Model Training

Based on the project requirements, four models need to be established separately and the performance between the models is compared. In order to facilitate observation and comparison, the following is the selection statement for the models:

Time window: Every Thursday in 2018 from 6 am to 9 am

Univariate model: independent variable Sub_metering_1

Multivariate model: Sub_metering_1, Sub_metering_2, Sub_metering_3

PCA univariate model: the first principal component (PC1) among the three variables of Sub_metering_1, Sub_metering_2, and Sub_metering_3

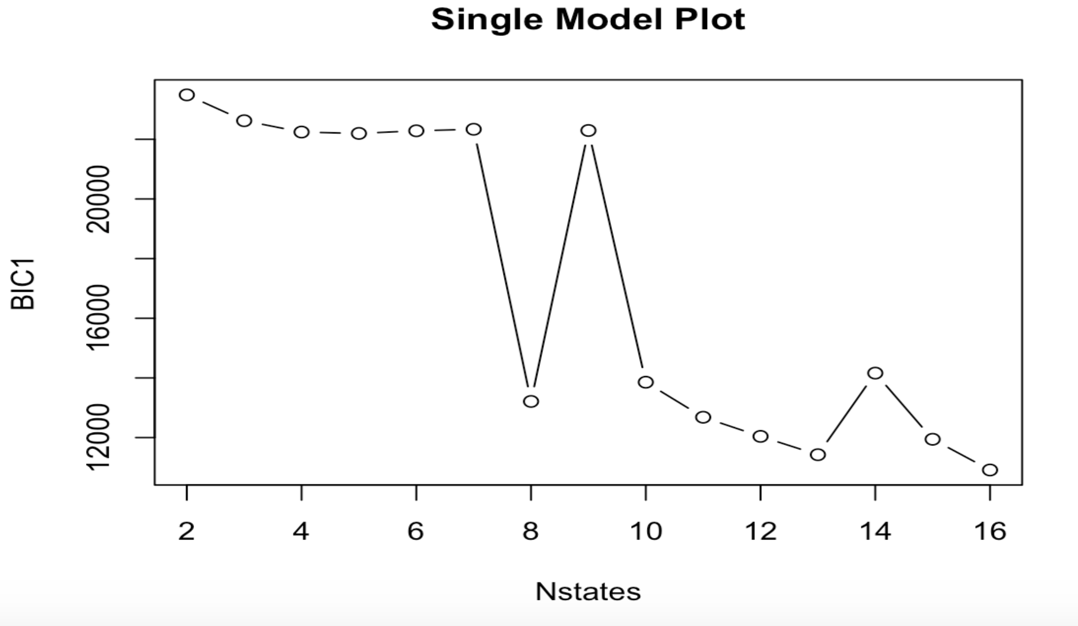
PCA multivariate model: the first two principal components of the three variables of Sub_metering_1, Sub_metering_2, Sub_metering_3 (PC1 & PC2)

To build four models of different situations, and at the same time, in order to create HMM models later, it is necessary to determine the best state number of each model. The state number of the HMM model is a hyper parameter, so a selection method should be adopted for the hyper parameter (similar to the number of categories in the cluster). The simplest method is to use the state value to try and use the BIC criterion to determine the best state number.

For this term project, it is the best choice to use the state value of 2-16 to test. The reason is that the starting point of the state value must be greater than 1, but the question is how many values must be reached to stop the test state number? Experiments have found that when the state number reaches 12, the overall function state gradually enters a fluctuating state, so it is reasonable to use 16 as the endpoint of the test state number. Choosing a larger numerical test will only increase the calculation time without any substantial significance.

The BIC criterion is to take the global minimum or local minimum. Take the following univariate model as an example:

The ordinate is called BIC1.



It is obvious that the minimum value is 16, but 16 cannot be selected as the best state number. Most of the state numbers need to choose a local minimum. The reason is that as the state value increases, the BIC drop will become smaller and smaller which is a fluctuating state. If the calculation counts toward the number of states to 30, it may get a lower result, but it is not an option. Generally speaking, looking for an inflection point or local minimum that quickly drops to a stable fluctuation can be used as the best state number.

In the figure above, the best state number as a univariate model is 13.

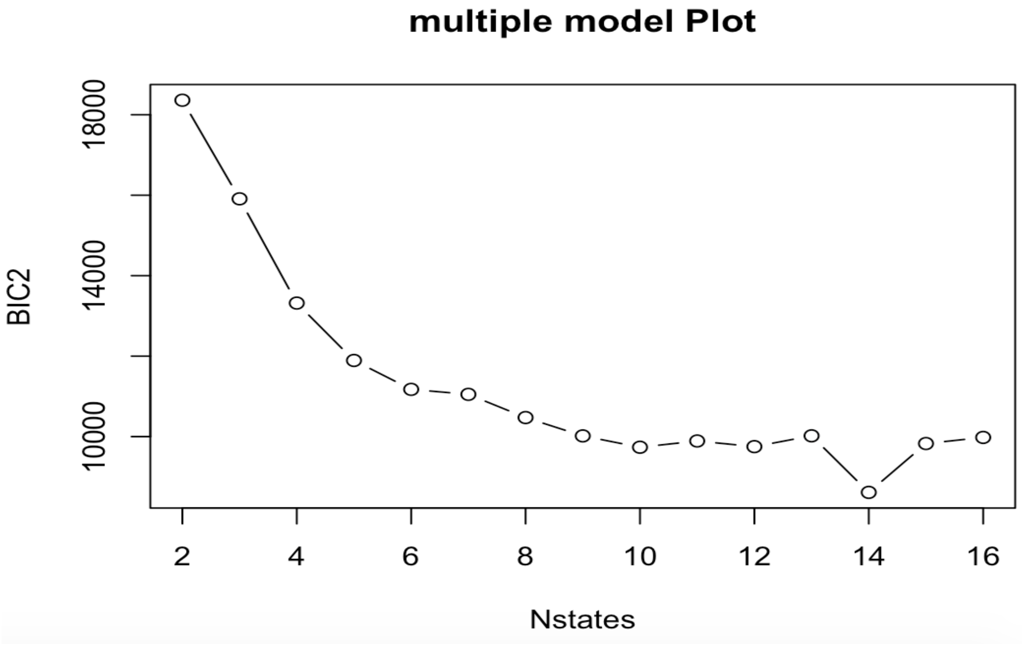
Output info:

```
Convergence info: 'maxit' iterations reached in EM without convergence.

'log Lik.' -4820.04 (df=194)

AIC:  10028.08

BIC:  11425.23
```

- -

Follow plot is shows the multivariate model：（Sub_metering_1、 Sub_metering_2、 Sub_metering_3）

The ordinate is called BIC2



The number of best states as a multivariate model is 12

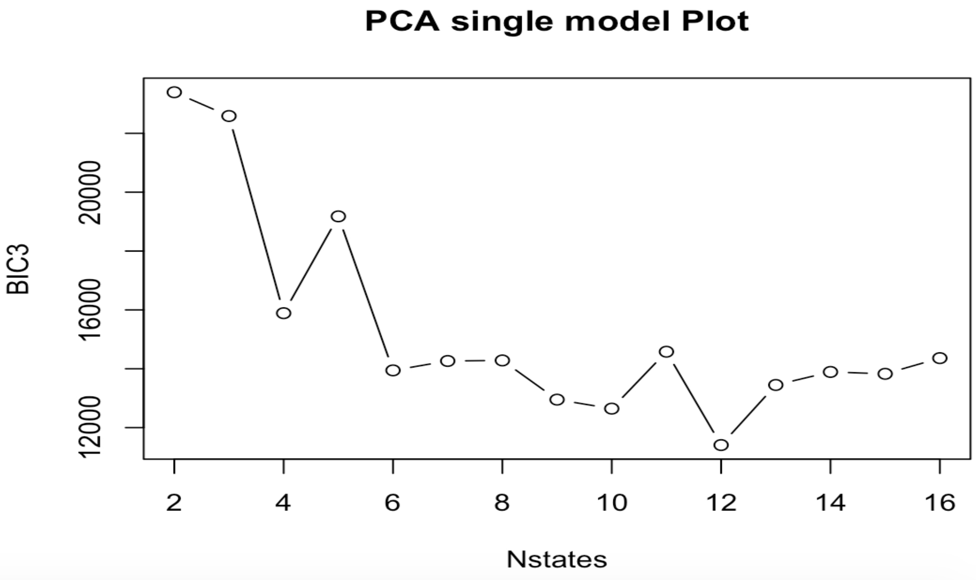
Output info:

```
Convergence info: 'maxit' iterations reached in EM without convergence.

'log Lik.' -3151.739 (df=251)

AIC:  6805.478

BIC:  8613.131
```

- -

The figure below shows the PCA univariate model: (the first principal component (PC1) among the three variables of Sub_metering_1, Sub_metering_2, Sub_metering_3)

The ordinate is called BIC3



The number of best states as a multivariate model is 12

Output info:

```
converged at iteration 479 with logLik: -4935.291
```

Convergence info: Log-likelihood converged to within tol. (relative change)

'log Lik.' -4935.291 (df=167)

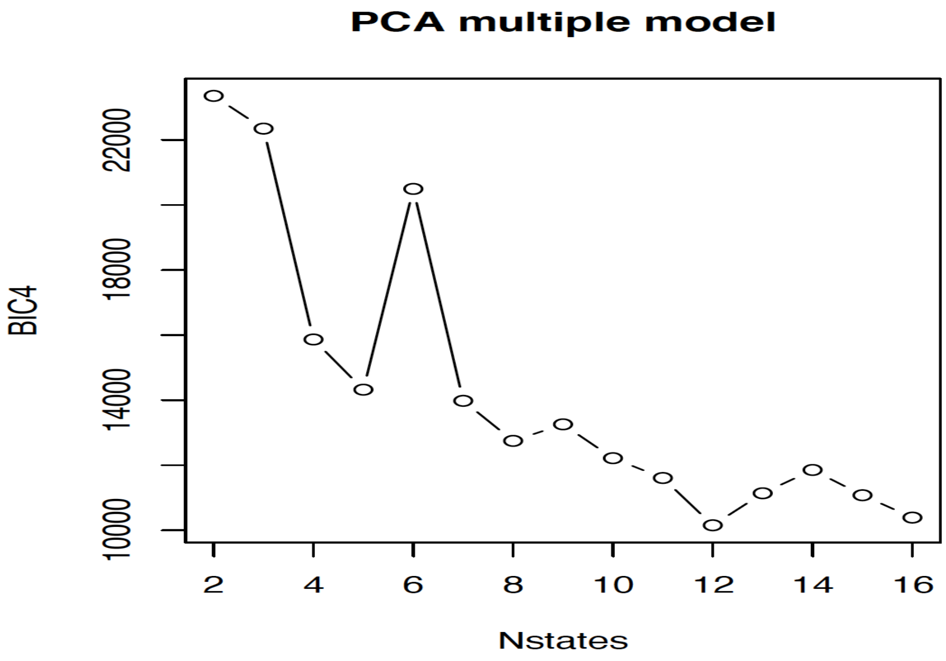
AIC: 10204.58

BIC: 11407.28

- -

The figure below shows the PCA multivariate model: (the first two principal components of the three variables of Sub_metering_1, Sub_metering_2, Sub_metering_3 (PC1 & PC2))

The ordinate is called BIC4



The optimal number of the model is 12

Output info:

Convergence info: 'maxit' iterations reached in EM without convergence.

```
'log Lik.' -4252.643 (df=179)

AIC:  8863.286

BIC:  10152.41

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

After finding the best state number for each situation, it is necessary to build an HMM model based on them. Use the depmix() and fix() functions to get the log-likelihood of the training set and experimental set for each situation respectively. Log-likelihood is an indicator to evaluate the predictive effect of the model, which is mainly used to see which of the models fits better. Generally speaking, the evaluation criterion is that the larger the log-likelihood value, the higher the fit of the model. Considering that most of the results are negative numbers when the standard is given, after taking the absolute number, the smaller the number represents the better the fit of the model. The following table is the test of the two data sets for the four models.

	Train Set Log-like	Test Set Log like
Single HMM Model	-4311.387	-1628.297
Multiple HMM Model	-4652.606	-1522.963
PCA Single Model	-5659.18	-2023.379
PCA Multiple Model	-5394.926	-1965.557

By using the horizontal comparison, it can clearly see that the overall fit of the test set is better than the fit of the training set. A reasonable explanation for this situation is that the training set accounts for 80% (8:2) of the total data, so it will be lower than the test set. However, comparing the four models longitudinally, since the test set is used to test the final effect of the model, it can be seen that the multivariate HMM model performs best. After the four models are finally determined through normal data, the three data sets with anomalies can be brought into the four models for testing. Similar to the above steps, by comparing the log-likehoods, compare the effects of abnormality and find the best model. The following table is a summary of the results of the four abnormal data:

	Anomaly data 1	Anomaly data 2	Anomaly data 3
Single HMM Model	-8460.15	-8971.28	-8788.997
Multiple HMM Model	-7699.624	-8188.738	-7951.568
PCA Single Model	-10108.19	-10410.95	-10081.88
PCA Multiple Model	-8512.811	-8970.858	-8396.03

1) By horizontal comparison, it shows that the fitting effects of each model on different abnormal data are different. It is not difficult to see that the anomaly degree of data2 is greater than data3, and the anomaly degree of data3 is greater than data1.

Anomaly data 2 > Anomaly data 3 > Anomaly data 1 (The degree of anomaly)

2) By longitudinal comparison,the PCA univariate HMM model has a lower degree of fit than the univariate HMM model, and the PCA multivariate HMM model has a lower degree of fit than the multivariate HMM model.

Single HMM model > PCA Single model > Multiple HMM model > PCA Multiple model

Although the univariate model is better than the PCA multivariate model, when observing the third abnormal data in anomaly detection, we can see that the effect of the PCA multivariate model is better than the univariate model. This shows that the data information is missing after the principal component extraction, and the PCA multivariate model's ability to detect this abnormal data has declined. But it is undeniable that PCA can greatly reduce the amount of data for model training and reduce model training time in the case of high-dimensional data.

Report Conclusion

Experience and lesson learned:

1) For the selected time window, the time period may be selected to be tested with the exception of the data collection time coincident. The original idea was that due to the possibility of external uncertain factors, normal data would be mistakenly identified as abnormal. For example, a bank card holder spends an average of $50 a day but spends $3,000 in the three days before and after Christmas. This data seems abnormal in daily life but under certain circumstances, it is normal data. Although the underlying laws of the data are consistent, the HMM model is originally derived. What can be improved is to adjust the time window so that it looks more rigorous and convincing.

2) It took some time to study the question of whether the same calculation result can be achieved by running the data model multiple times. In fact, because one of the collected samples uses random sampling, there will be differences due to machine problems and the number of convergences during training, so it is very difficult to achieve the same training results.

Develop Learning (Reinforcement Learning)

The general database will update the data at an extremely fast speed and the large number of data is difficult to carefully observe the existence of abnormalities. However, intruders generally use highly concealed methods to hide into seemingly "normal" data. It is difficult for humans to distinguish and make correct judgments quickly. In reality, the reason why the network can be relatively safe depends entirely on systematic reinforcement learning theory. Next section introduces in detail how reinforcement learning plays a role in analyzing abnormalities from analysis principles to examples.

Introduction to Reinforcement Learning and its Application in Anomaly Detection

Group 14

Richard Li, Alex Cheung

Simon Fraser University CMPT318

2020/11/29

Introduction

In March 2016, AlphaGo, developed by DeepMind which is a subsidiary of Google, successfully defeated the world Go champion Lee Se-dol by a score of three to one, and again defeated the world Go champion Ke Jie in May of the following year. So far, the view that AlphaGo's chess power has surpassed the top level of human professional Go has become an undeniable fact. It makes people feel the power of artificial intelligence (AI). The AlphaGo is so magical thanks to working under a principle called "reinforcement learning". Reinforcement learning seems to be very far away from daily life, but it protects people's property and information security in the era of highly developed information. The Internet has now covered the whole world, greatly increasing the scale of system coverage, and at the same time it has brought about a more dangerous and fragile network security environment more than ever before. The complexity and uncertainty of network attacks requires the protection mechanism to be responsive, scalable and self-adaptive. Statistical research shows that 62% of attacks are recognized after they cause major damage to the network system. Therefore, with the continuous acceleration of the information age and the continuous escalation of attack methods, how can the surveillance and defence system grow with the times? This is inseparable from the help of reinforcement learning. This article focuses on the concept of reinforcement learning, through some simple examples to tell what reinforcement learning is composed of, what are the uses of the components, and how they work together. Moreover, it combines multiple examples to show the importance of reinforcement learning as an online machine learning method for network intrusion detection and what specific help it has for the future well-being of mankind.

Why is reinforcement learning needed?

Before specifically explaining reinforcement learning, two related terminologies are needed to be introduced, supervised learning and unsupervised learning. Reinforcement learning, supervised learning, and unsupervised learning are the most important branches of the field called machine learning(ML). Supervised learning has a label, which indicates to the algorithm about what kind of input should be for what kind of output. Unsupervised learning does not have a label. It can be seen that the overall supervised learning or the relatively rigid input-output corresponding model cannot provide the flexibility which is necessarily needed to deal with various scenarios. However, the main difference between supervised learning and reinforcement learning is whether the feedback received is evaluative or instructive. Instructive feedback is to provide methods to achieve goals, and supervised learning is to solve problems based on instructive feedback. Reinforcement learning uses evaluative feedback, which provides information based on how far the goal will be achieved. For example, for a system that controls oxygen content, the operator cannot tell the algorithm what the correct setting of each part is at any given time. Such instructive feedback does not seem to have any effect. However, the staff can easily obtain the various instrument data generated in a specific period. This kind of evaluative feedback idea is more intuitive and easier to realize. To sum up, reinforcement learning is a flexible learning method in machine learning. Due to the four factors behind it, which can cope with more complex and changeable environments.

What is reinforcement learning?

Reinforcement learning is considered to be the closest form of human learning because it can learn based on its own experience by exploring unknown environments. It includes four main factors: agent, status, reward, and action. To facilitate understanding, take the robot walking a maze as a scene to illustrate what agent, status and reward factors represent. An agent is a hypothetical entity, usually an object trained in a specific environment to make correct decisions. In the example, the agent can be understood as a robot, what it has to do is try to get out of the maze without any collision. The state defines the current real-time state information of the Agent, such as the robot's position in the maze, movement posture, current movement speed and body distortion angle. The state information of the agent entirely depends on the method of solving the problem. When an Agent performs a specific action or task, it will receive real-time feedback called a reward, which will be treated as a scalar. Based on the execution of the behaviour in the current environment, the reward can be divided into positive rewards and negative rewards. For example, as the absolute distance of the robot from the exit is greater, the score will drop, (Bad reward) vice versa. At this point, having the four elements of the agent, state, action, and reward can form a simple Markov decision process, which is also the main theory behind reinforcement learning.

The Fundamental Principles of the Markov Decision Process

Many previous states need to be taken into account in the transition of the real environment, which is complicated and difficult to model. This can be simplified by assuming that the state is converted to Markov property (Markov property refers to the probability of transition to the next state only related to the current state, and not related to any previous state). The dynamic process of MDP is as follows. The initial state of an agent is S_0, and an action a_0 is selected from the actions for execution. After execution, it randomly transfers to the next state S_1 according to the state transition probability PS_0, Pa_0. The following figure shows a sketch of the processes.

$$S_0 \xrightarrow{a_0} S_1 \xrightarrow{a_1} S_2 \xrightarrow{a_2} S_3 \ldots$$

Figure 1.1 MDP process

After each decision-making will receive the corresponding reward, and the goal of reinforcement learning is how to gradually form an expectation of the stimulus under the incentives or punishments given by the environment, to find the optimal strategy to maximize the long-term future reward. The following two pictures are the goal of reinforcement learning by David using the maze as an example. (As just mentioned, when the absolute distance of the robot from the exit is greater, the score will be -1).

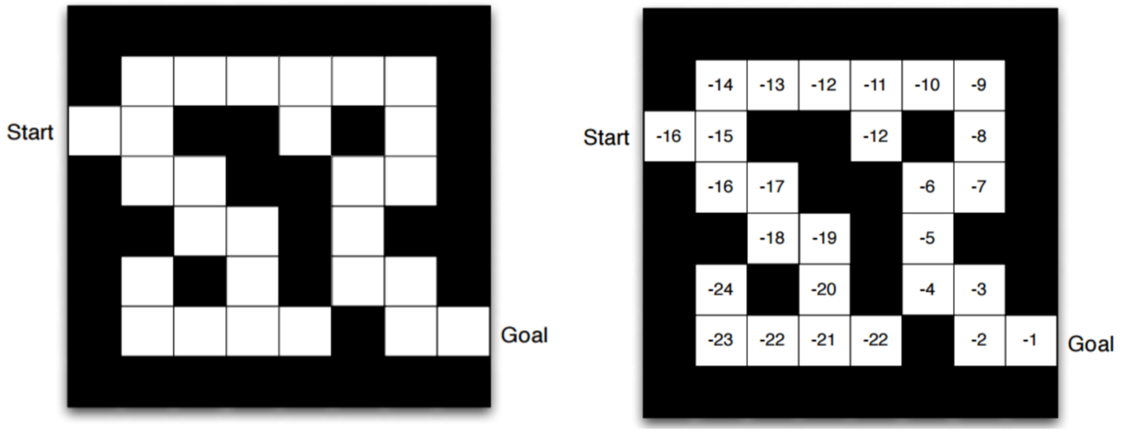


Figure 1.2 Example of RL (Value Iteration)

By illustrating some basic examples, we already know the basic principles of reinforcement learning. Although these jobs can be done by humans even faster than computers can run. So what difficult tasks can be done by reinforcement learning instead of humans is a question. In other words, what benefits reinforcement learning can bring to human well-being needs to be explored.

Reinforcement Application in Real Life

Reinforcement learning can also be used in the self-driving car industry. Some of the autonomous driving tasks, such as trajectory optimization, motion planning, dynamic pathing, controller optimization, and scenario-based learning policies for highways, can all be achieved by installing cameras on cars and applying reinforcement learning. Based on the camera captures of states of cars speed, driving lanes, and location, the system makes actions and gets corresponding rewards for tasks. As more reinforcement learning is processed, autopilot systems will be better able to use faster than the human brain to cope with more complex road conditions. This can improve the efficiency, intelligence and safety of self-driving cars. The more varieties of qualified disabled people can drive cars as normal people do. Reinforcement learning can also be applied in automated medical diagnosis, chronic disease and critical care systems. The system will evaluate the patient's states of diseases and give a stable and reasonable solution for the specific states of patients. The use of RL in healthcare enables the improvement of long-term outcomes by factoring in the delayed effects of treatments. For example, [1]KenSci uses reinforcement learning to predict patients' dynamic changes and help practitioners to find treatments at patients' early stages.

Reinforcement learning applied in anomaly detection

As the volume of data surges rapidly when the network size is enlarged. Using manual network data anomaly detection is impossible. Reinforcement learning plays a huge role in finding anomalies for network intrusion detection in the network traffic flow. A network intrusion detection system is a software or hardware platform installed on network equipment to detect and report to the administrator abnormal or malicious activities by analyzing the audit data. The architecture can be implemented by setting network flow parameters as the state variables, such as the port number, packet size and transmission rate. The action can either flag or not flag an anomaly detection warning based on Q(s, a) values given by the output of the neural network in the previous state and an ε-greedy manner. The following image is the network intrusion detection architecture.

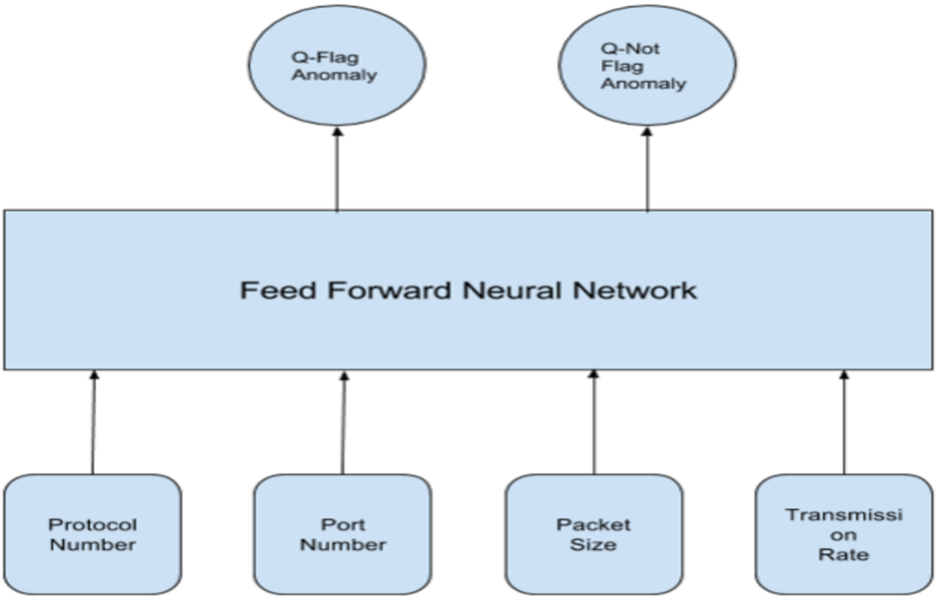


Figure 1.3 Network Intrusion Detection Architecture.

The number of reinforcement learning processes for network intrusion detection could be generalized into 4 parts of the back propagation algorithm. [2] The first is to initialize all the weights in the deep neural network with random values and Initialize the total accumulated reward to zero. And then to get an initial state from the environment. A third part is a finite number of recursive steps which contains starting with the state obtained in the previous step; performing a feed-forward of the current state using deep neural network, and getting the predicted Q(s, a) values; Taking action of either flag or not flag from the current state, based on the Q(s, a) values given by the output of the deep neural network in the previous state and an ε-greedy manner; Getting the reward and moving to state; Passing the new state into the deep neural network and calculate the Q(s, a) values by Bellman's equation; Performing a training of the deep neural network by back propagation of the error of prediction, where the difference between target Q(s, a) and predicted Q(s, a) is taken as the error of prediction. The last part is computing the new cumulative total reward. Through this advanced and comprehensive network anomaly detection reinforcement learning process, hackers can be identified, such as Denial-of-Service (DoS) attack, Remote-to-Local (R2L) attack, User-to-Root (U2R) attack and the Probing attack. This can prevent people from being stolen and being accessed private data unauthorized which guarantees to access the network under a private and trustworthy environment for everyone.

Conclusion

In summary, reinforcement learning is a concept of learning while obtaining examples based on Markov Decision. The learning system is formed through agent, reward, state and action to find a way to maximize future returns. Reinforcement learning technology has been widely used in many fields, such as telecommunications, manufacturing, power management, healthcare, government, and even entertainment. With the advent of the technological age, everything is closely related to the Internet. As Thanh Thi Nguyen and Vijay Janapa Reddi said, "The scale of Internet-connected systems has increased considerably, and these systems are being exposed to cyber-attacks more than ever." (Nguyen & Reddi, 2019, p. 1). For network security, traditional supervised learning (learning through data labels) is far from sufficient to resist network attacks. The extremely fast adaptability of reinforcement learning can perfectly cope with the complex and changeable characteristics of network systems. Even so, the speed of network update is amazing, and the variety of attack methods and the degree of concealment have also increased. Developers must continuously improve the functionality and anomaly awareness of reinforcement learning to deal with various threats.

References

AlphaGo. (2018, December 18). Retrieved December 1, 2020, from Deepmind.com website: https://deepmind.com/research/case-studies/alphago-the-story-so-far

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Great Learning Team. (2020, February 17). Use of Reinforcement Learning (RL) in healthcare. Retrieved December 1, 2020, from Mygreatlearning.com website: https://www.mygreatlearning.com/blog/reinforcement-learning-in-healthcare/

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., & Pérez, P. (2020). Deep reinforcement learning for autonomous driving: A survey. Retrieved from http://arxiv.org/abs/2002.00444

Koduvely, D. H. (2018, January 19). Anomaly detection through reinforcement learning. Retrieved November 30, 2020, from Zighra.com website: https://zighra.com/blogs/anomaly-detection-through-reinforcement-learning/

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979) *Multivariate Analysis*, London: Academic Press.

Nguyen, T. T., & Reddi, V. J. (2019). Deep reinforcement learning for cybersecurity. Retrieved from http://arxiv.org/abs/1906.05799

Rungta, K. (2020, January 1). Reinforcement Learning: What is, Algorithms, Applications, Example. Retrieved December 1, 2020, from Guru99.com website: https://www.guru99.com/reinforcement-learning-tutorial.html

Sharma, A., Kalbarczyk, Z., Barlow, J., & Iyer, R. (2011). Analysis of security data from a large computing organization. 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN). IEEE.

Venables, W. N. and B. D. Ripley (2002) *Modern Applied Statistics with S*, Springer-Verlag.Ripley, B. D. (1987) *Stochastic Simulation*. Wiley.

Vincent, J. (2019, November 27). Former Go champion beaten by DeepMind retires after declaring AI invincible. Retrieved December 1, 2020, from The Verge website: https://www.theverge.com/2019/11/27/20985260/ai-go-alphago-lee-se-dol-retired-deepmind-defeat

Report Reference

https://builtin.com/data-science/step-step-explanation-principal-component-analysis
https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp
https://cmdlinetips.com/2019/04/introduction-to-pca-with-r-using-prcomp/

Training and Test Sets: Splitting Data. (2020.2.10). Retrieved December 3, 2020, from Google.com

https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data

Shah, T. (2017, December 6). About Train, Validation and Test Sets in Machine Learning. Retrieved December 3, 2020, from Towards Data Science website: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/sample
http://www.rexamples.com/14/Sample()
https://www.rdocumentation.org/packages/compositions/versions/2.0-0/topics/runif
https://www.programmingr.com/examples/neat-tricks/sample-r-function/r-runif-uniform-distribution/
Wikipedia contributors. (2020, October 23). Hyperparameter (machine learning). Retrieved December 3, 2020, from Wikipedia, The Free Encyclopedia website: https://en.wikipedia.org/w/index.php?title=Hyperparameter_(machine_learning)&oldid=984957886
https://my.oschina.net/u/876354/blog/1614879
https://ishbel.host.cs.st-andrews.ac.uk/WhatisaTechnicalEssay.pdf

Brownlee, J. (2017, July 25). What is the Difference Between a Parameter and a Hyperparameter? Retrieved December 3, 2020, from Machinelearningmastery.com website: https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/