Richard

Hi, I’m Junchen from Gorup 14

(ALex)This is Alex from Group 14 as well.

 We are presenting on data analysis of power consumption using the Principal Component Analysis (PCA).

The term project is focused on how we completed this research, what difficulties we met, how we solved the problems, and what we learned.

Our project is divided into four parts. We need to understand the given data, know how to work with the data, and find and create four Hidden Markov Models (HMM) for anomaly detection based on normal data set by using PCA. We use the HMM models to calculate and compare log-likelihood and test the anomaly data by comparing the log-likelihood for analyzing the anomaly situation.

First, we did preparatory work for why we are using the 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. So what exactly is the PCA doing?

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 represent the characteristics of the original data set. 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. That’s the reason why PCA is useful for analyzing data.

Alex

After knowing the main tool for this project, we determined the time window, we chose the time of every Thursday from 6 am to 9 pm. This is the screenshot of our data for an example. During these nine variables, we need to focus on what happens to the global active power. This image displayed here gives an overview of how the average global active power looks like. As I mentioned before, there are nine variables, so which variables will most affect the

change in global active power?

Based on our first assignment, we calculated the correlation among those 9 and got 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. We notice the Global\_intensity is closest to +1, so can we choose it?

The answer is no. 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.

Also, 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.

We can see the second level of correlation such as the sub\_metering\_1/2/3. These three are the best choices for PCA and will be our main research objects of experiment.

Richard

Next, we use PCA to figure out the number of factors. We use the procomp() to analyze the components after choosing the PCA variables. The image here is the information on using the summary function on PCA. So here, 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, we can say the dimension of the data has been successfully changed from 12480x13 to 12480x2. By the way, there are totally 12480 items in our data set.

However, when we received the result, we found that the variance of PC1 and PC2 was too large, especially since the variance of PC1 was as high as 6. We are unsure why the variance was so high. 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. It means in order to decrease both the mean and variance to be 1 before PCA analysis. This action is also known as standardization. After standardization, the summary looks like \_\_\_\_. As seen in the figure, the cumulative proportion of the first principal component accounts for 66.69%, and for 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. The potential threat has now disappeared.

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. But what's the ratio of dividing the sets? Well, The Machine Learning Crash Course seems to give us an answer: "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.” We 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.

Alex

There is a problem with the use of data classification functions for training and testing. We use the sample function for random sampling in assignment 3, 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 called 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. So we did a little bit upgrade for our code. Until now, we can have all sources for creating HMM models in four scenarios.

As the review, the Univariate model is using sub-metering-1 ; Multivariate model using all three Sub\_metering-1/2/3. PCA univariate model uses the first principal component ( we called it PC1) among the three variables of Sub\_metering\_1/2/3. PCA multivariate model will use the first two principal components. Remember only two components will cover all three parts ( we call them PC1 & PC2).

In order to build four models of different situations, it is necessary to determine the best state number of each model. The state number of the HMM model is a hyper parameter, so the simplest method is to use the state value to try and use the BIC criterion to determine the best state number.

 We know that the starting state value must be greater than 1, but the question is which value should be the stop state?

Richard

After some experiments on state values, we have found that when the state number reaches 12, the overall function state gradually enters a fluctuating state, so we thought it is reasonable to use 16 as the endpoint for the test state. If we choose a larger numerical test number, it will only increase the calculation time without any significant change.

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 we can???, we can calculate it toward the number of states to 30, you may get a lower result, but it is not an option. It will be better to look for an inflection point or local minimum number that quickly drops to a stable fluctuation will be used as the best state number.

So we found these plot have these nstate number respectively. So Yes, we found the Optimum state number, now 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 test set for each situation respectively. ???

Quick point here, 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.

Alex

The following table is the test results of the two data sets for the four models. Compared the four models horizontally, 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. So, right now we just keep the image and we will use it to compare later. 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-likelihoods, compare the effects of abnormality and find the best model. This table is a summary of the results of the four abnormal data: let’s analyze the result to see what can we get in it. By horizontal comparison, 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. From the degree of anomaly.

By vertically comparison, the PCA single HMM model has a lower degree of fit than the single HMM model, and the PCA multivariate HMM model has a lower degree of fit than the multivariate HMM model.

We can call back the previous table and see here. Although the single 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 single model. This shows that the data information is missing after the PCA, and the PCA multiple 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.

 Richard

The above is all the work that we have done for the data anomaly analysis topic. The main idea is we want to find an appropriate data model that can represent "normal" system behavior as much as possible without the actual situation. We also know how the PCA works. As the network advances in daily life, the database will update the data at an extremely fast speed. The large number of data is difficult to carefully observe the existence of abnormalities. Intruders can use highly concealed methods to hide into seemingly "normal" data. It will be hard for us 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.  We do a little bit of extending learning talking about what Reinforcement Learning (RL) is , what it consists of, how it works, and what applications it can be applied to.

There are 3 important types of Machine Learning. They are Supervised Learning, Unsupervised Learning and Reinforcement Learning. Supervised learning has labels that indicate to the algorithm what input should be for that output, while unsupervised learning has no labels. The different feedback they received shows that supervised learning is not as flexible as reinforcement learning. This is why reinforcement learning is popular. So what exactly is reinforcement learning?

Reinforcement learning is considered to be the closest form of human learning. Because it can learn by itself, based on its own experience by exploring unknown environments. It includes four main factors: agent, status, reward, and action. It will be easy to understand if we take the robot walking a maze as an example.

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 MDP, which is also the main theory behind reinforcement learning.

 Many previous states need to be taken into account in the transition of the real environment, which is 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). So the dynamic process of MDP is as follows. The initial state of an agent is S0, and an action a0 is selected from the actions for execution. After execution, according to the state transition probability PS0, Pa0, it randomly transfers to the next state S1. So that is the process of MDP. Remember the maze example, we need to add reward in it. So, after each decision-making will receive the corresponding reward, as we can see photos here, although it is a example of value iteration, the point is the goal of RL 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.

Alex

 Reinforcement learning real life application examples.

Reinforcement learning can be used in the self-driving car industry. Some auto driving tasks, like motion planning, dynamic pathing and scenario-based learning policies for highways, can all be achieved by applying reinforcement learning via car cameras. Based on captures of states of cars speed, driving lanes, and location, the driving system makes actions and gets corresponding rewards for tasks. As more reinforcement learning is processed, autopilot systems will be able to immediately give solutions to more complex road conditions. This can improve the efficiency, intelligence and safety of self- driving cars. (Tesla)

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 the patient. Using Reinforcement Learning in healthcare improves long-term outcomes by factoring in the delayed effects of treatments. For example, KenSci uses reinforcement learning to predict patients' dynamic changes. Based on the RL action-reward memory, help doctors to find the best treatments at patients' early stage. (healthcare)

As the volume of data increases rapidly when the network size gets larger. It’s impossible to manually detect anomaly data in the network. Now,  Reinforcement Learning can play an important role in finding anomalies for network intrusion detection in the network traffic flow.

The network intrusion detection system is a software or hardware platform that is installed on network equipment to detect and report to the administrator with abnormal or malicious activities by analyzing the audit data.

The image below is an example of intrusion detection architecture.

We can set the 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 Quality function Q(s, a) values.

Quality function Q(s, a) which gives an estimate of the maximum total reward or payoff the agent can receive starting from state S and performing action A. The value of Q(s, a) for all states and actions can be found through solving the Bellman equation.

the process can be :

first to Initialize all the weights in Deep Neural Network with random values and Initialize the total accumulated reward to zero.

Then, Get an initial state from the environment and get into recursive steps.

Recursive steps start with the initial state or start with the state obtained from the previous step. And perform a feed forward of the current state by using Deep Neural Network, and get the predicted quality function Q(s, a) values.

Then, it will take an action of either flag or not flag from the current state, based on the quality function Q(s, a) values given by the output of the Deep Neural Network in the previous state.

Then, Get the reward and the next state from the environment.

* 1. Pass the new state also through the Deep Neural Network, to compute the target quality function Q(s, a) values using Bellman's equation.

And next, perform the training of the DNN by back propagation of the error of prediction, where the difference between target Q(s, a) and predicted Q(s, a) in the previous step is taken as the error of prediction.

And finally, Compute the new cumulative total reward.

Once enough recursive steps are completed, the Deep Neural Network would have adjusted its weights to predict correct Q(s, a) values, so that the agent would be able to act with optimal policies in any given state and take correct actions of flagging malicious network connection requests.

The lessons that we have learned from the term project experience.

For the term project, 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 it can be treated as normal data under certain circumstances. 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.

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.

That's all for our presentation!

 Since we only have 2 people in our group, the quality of presentation may not be the best.

Thanks for your patience and listening to our whole presentation.