Point Process Course Project

Project Introduction

This project aims to improve the understanding of the point process method based on neural network by using neural point process to solve practical problems. This project contains two sub-tasks involving Event Recurrent Point Process (ERPP) and Recurrent Marked Temporal Point Process (RMTPP).

Project Background

This project is derived from Predictive Maintenance Problem. It involves equipment risk prediction to allow for proactive scheduling of corrective maintenance. Such an early identification of potential concerns helps deploy limited resources more efficiently and cost effectively, reduce operations costs and maximize equipment uptime. Predictive maintenance is adopted in a wide variety of applications such as fire inspection, data center and electrical grid management. For its practical importance in different scenarios and relative rich event data for modeling, this project is modeling a real-world dataset of more than 1,000 automated teller machines (ATMs) from a global bank headquartered in North America.

Traditional models either suffer from model misspecification if the chosen parametric intensity function does not fit with the real behavior of the event data, or the learning algorithm can be mathematically very complex for nonparametric models. With the fast development of deep learning theory and techniques, especially for recurrent neural network models, there is a trend for adapting networks to temporal point process learning. Neural point process frees the need for explicit parametric intensify form selection, thus the neural point process models shows high model capacity for learning arbitrary and unknown distributions.

We have no prior knowledge on the dynamics of the complex system and the task can involve arbitrarily working schedules and heterogeneous mix of conditions. It takes

much cost or even impractical to devise specialized models. In this scenario, traditional parametric point process is difficult to model event sequences, while neural point process which has higher capacity to digest massive event data can achieve better results. In the experiment, you can try to compare the performance gap between traditional parametric point process and neural point process.

Dataset

In maintenance support services, when a device fails, the equipment owner raises a maintenance service ticket and technician will be assigned to repair the failure. The studied dataset is comprised of the event logs involving error reporting and failure tickets, which is originally collected from 1,554 ATMs. The event log of error records includes device identity, timestamp, message content, priority, code, and action. A ticket (TIKT) means that maintenance will be conducted. A ticket (TIKT) means that maintenance will be conducted. Statistics of the data is presented in Table1. The error type indicates which component encounters an error:

- printer (PRT)
- cash dispenser module (CNG)
- Internet data center (IDC)
- communication part (COMM)
- printer monitor (LMTP)
- miscellaneous e.g., hip card module, usb (MISC)

The training data consists of 1085 ATMs and testing data has 469 ATMs, in total Wincor ATMs that cover 5 ATM machine models: ProCash 2100 RL (980, 430), 1500 RL (19, 5), 2100 FL (53, 21), 1500 FL (26, 10), and 2250XE RL (7, 3). The numbers in the bracket indicate the number of machines for training and testing.

In dataset, event types have been numerically, and the specific correspondence is shown in the table 2.

Table 1 ATM data set different types of event statistics table

Type	total
TIKT	3516
PRT	99252
CNG	80921
IDC	15074
COMM	165567
LMTP	49094
MISC	138791

Table 2 ATM event type correspondence relation

Type	Number
TIKT	6
PRT	0
CNG	1
IDC	2
COMM	3
LMTP	4
MISC	5

Dataset contains two data files：train.csv and test.csv

In data files, the following data is provided:

- id: ATM code
- time: timestamp
- event: event type

Sampling

You can choose the number of historical events considered for each event, such as 10 or 15. According to this strategy, multiple sequences can be sampled from each ATM sequence.

Tasks

Figure 1 shows the model architect. For a given sequence $S = ((t_j, y_j)_{j=1}^{n})$, at the j-th event, the marker $y_j$is first embedded into a latent space. Then, the embedded vector and the temporal features are fed into the recurrent layer. The recurrent layer learns a representation that summaries the nonlinear dependency over the previous

events. Based on the learned representation h_j, it outputs the prediction for the next marker $\hat{y}_{j+1}$and timing $\hat{t}_{j+1}$ to calculate the re- spective loss functions.

Task1 ERPP

Model ERPP is Event (sequence) hRNN mentioned in reference paper. It just reduces time series RNN. More details to view the paper.

Reference paper: S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu. Modeling the intensity function of point process via recurrent neural networks. In AAAI, 2017.

Task2 RMTPP

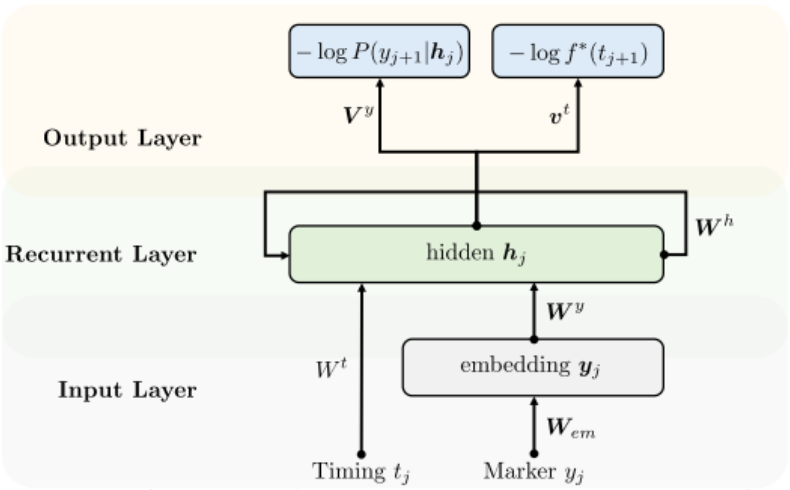
Compared to ERPP, RMTPP construct a semi-parametric point process, utilizing intensity function to predict timestamp. More details to view the paper.

Reference paper: N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vectore. In KDD, 2016.

Evaluation metrics

- Event Type Prediction：Precision, Recall, macro-F1 Score and Confusion matrix over 2 main types (‘error’, ‘ticket’) as well as Confusion matrix over 6 subtypes under ‘error’. Note all these metrics are computed for each type, and then are averaged over all types.
- Event Time Prediction: Mean Absolute Error (MAE) which measures the absolute difference between the predicted time point and the actual one. These

Figure 1 Model Architect



Environment Configuration

It is recommended to perform tasks in Python environment. You should install necessary libraries and process the datasets. Try to use parallel computing, whether you use CPU or GPU.

All tasks do not limit the models and methods to be used. For beginners, you can start with the installation of anaconda and build a basic Python environment. Numpy and Pandas are mainstream tools for algorithm implementation and data processing. Numpy supports accelerated vector operations, which is more efficient than using a 'for' loop in Python. We recommend installing tensorflow or pytorch, which has a CPU version and a GPU version. There are also many environment management tools, 'conda' on anaconda, git on github and pip.

Project Submission

In project report, you should describe the two tasks separately. For each task, description of the environment configuration, solution, and test results must be included. In environment configuration part, you should describe the software language, key libraries, and other things necessary to run the program. In solution part, you should explain what kind of problems are encountered and how to solve them. In test results part, you should display the actual results of the program and analyze them.

You should submit a .zip/.rar file with all source code and the report.

Academic misconduct and handling

This project has tools to analyze the plagiarism of source code and reports, and don't try to test its reliability. For self-completed student, there will be reward points. For plagiarized student, it will be awarded zero points. If you are convicted of plagiarism, you can ask TAs to make a defense.

Questions

First, Google or Baidu it. It's a good habit to use the internet to answer your questions. For 99% of all questions, the answer is easier found online than asking us. Also make sure to check out the helper resources we have linked above. If you can't figure it out this way, post your question on Piazza (preferably public question rather than a private one, so that everyone can benefit from the given answer) or send us an e-mail.