Research on Machine Learning Algorithms and Feature Extraction for Time Series

Lei Li*, Yabin Wu*, Yihang Ou*, Qi Li*, Yanquan Zhou* and Daoxin Chen†
*School of Computer
Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Beijing, China, 100876
Email: leili@bupt.edu.cn
†CapInfo Company Limited, No.23Zhichun Road, Haidian District, Beijing, China, 100191
Email: chendaoxin@capinfo.com.cn

***Abstract*—This paper aims to use various machine learning algorithms and explore the influence between different algorithms and multi-feature in the time series. The real consumption records constitute the time series as the research object. We extract consumption mark, frequency and other features. Moreover, we utilize support vector machine (SVM), long short-term memory (LSTM) and other algorithms to predict the user's consumption behavior. Besides, we have also implemented multi-feature fusion and multi-algorithm fusion with LSTM and SVM. Eventually, the experimental results show that LSTM algorithms is advantageous in prediction when the data is sparse. In the other hand, the SVM is beneficial when the data is more abundant. What's more, LSTM-SVM fusion model has advantages on the extracting features of LSTM and on the classification of SVM. In most cases, LSTM-SVM is most outstanding in prediction.**

I. Introduction

With the rapid development of Internet and information technology, massive data is emerging. Therefore, data mining becomes one of the most important research fields nowadays. Time series analysis is a method to explore all the information contained in the time series, to observe, to estimate and to study the statistical regularity in the process of long-term change in such a set of real data [1]. The combination of time series and data mining can explore the changing laws of phenomena and make predictive control of the actions that have not happened.

As early as 1927, the British statistician Yule put forward the Autoregressive (AR) model [2] used to predict the law of market changes. In 1931, Walker established the Moving Average (MA) model. Later, he combined these two models to establish an Autoregressive Moving Average (ARMA) model [3]. Based on previous research, Box and Jenkins proposed the autoregressive integrated moving average (ARIMA) model which is of great significance for modern time series analysis and prediction [4]. These models are known as time series predictive studies of classical methods.

978-1-5386-3531-5/17/$31.00

With the recent rapid development and extensive use of machine learning and neural networks, their combination with time series data mining has become a hot issue. Thissen et al. [5] use ARIMA, Support Vector Machine(SVM) and the Recurrent Neural Networks (RNN) model to predict on different time series data sets and to compare the effects of the various models under different tasks. Tian et al. [6] and Fu et al. [7] use Long Short-Term Memory (LSTM) neutral network to achieve traffic flow forecast. Wang et al. [8] use LSTM for earthquake prediction. Junxiang et al. [9] use RNN to construct a framework about space-time forecasting for the time series formed by air pollutants. Kim [10] uses SVM to predict the financial time series.

From above studies, there are few studies focus on the effect of time series feature engineering on the prediction effect of various algorithms. Thus we plan to work on the influence of various features in different algorithms for the time series, including the representative SVM and LSTM algorithms. Through multi-feature integration, we analyze the effect of the features. In addition, we combine the two algorithms of LSTM and SVM. Experimental results show that it can combine the memory advantage of LSTM and the classification advantage of SVM, and obtain the best prediction result.

II. System Design

A. System architecture

Figure 1 shows our system architecture.

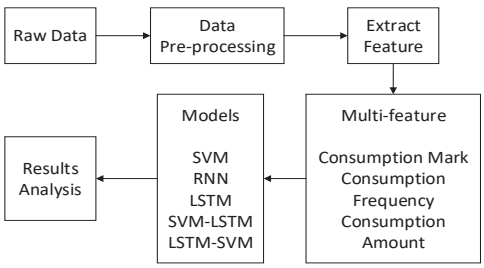

Fig. 1. System architecture

B. Raw data

Our raw data is collected from the real users' consumption records, containing 21340966 records of 538052 users. The consumption time span of the data is from June 26, 2014 to November 10, 2016, a total of 869 days. We randomly select 50000 users, and extract their consumption records as the data set for our experimentation.

C. Pre-processing

We extract all records, implement data cleaning and integration, then saving the required features.

D. Feature extraction

We divide 869 days' data into 124 weeks which means time series include 124 dimensions. Then we extract the following features in this time series.

Consumption mark: It refers to whether the user consume within a week, if consumption occurs then recording as 1, otherwise, recording it as 0.

Consumption frequency: It is the count of each users consumption times per week.

Consumption amount: It is the number of one's all consumption money within a week.

E. Multi-feature fusion

The above features can reflect the user's consumption behavior from different angles. In order to support more accurate knowledge mining and prediction, it is necessary to integrate a variety of features, and complement each other. So we try to combine the above features in different ways in our experiments.

F. Model Algorithms

As mentioned earlier, this paper focuses on machine learning and deep learning algorithms, hence the following three algorithms are chosen for research.

1) SVM: SVM is a supervised learning model whose essence is a binary-classification model. Its basic model is defined in the feature space on the largest linear classifier [11].

2) RNN: Compared with the traditional neural network, RNN has added a cyclic structure, which can keep the persistent memory of information.

3) LSTM: LSTM is a variant of RNN. LSTM achieves the purpose of maintaining the persistence of information through the control of the "gate" inside the neuron, and avoids the problem of long-term dependence in RNN.

In our experiments, for all network structures, the number of hidden units is 128.

4) Integration of SVM and LSTM: Cimino et al. [12] propose a series of LSTM-SVM structures for feature extraction and classification, and effective in subject classification and irony detection. In this paper, we try to combine SVM with LSTM in series, using the results of SVM as the input to LSTM and the results of LSTM as the output, which is called SVM-LSTM. Similarly, we exchange the position of SVM and LSTM, then we form the LSTM-SVM.

G. Evaluation method

The task of our experimental system is mainly to predict whether the user will consume or not in the next time unit, hence, we will pay more attention to the precision of the forecast. According to the confusion matrix [13], the precision P is defined as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

In addition, in order to accurately describe the rich degree of the data, we define a data abundance index K. Suppose that a user group contains N users, each of which has consumption records for Ci (i = 1, 2, ..., N) weeks in 124 weeks. This user group data has the rich degree as follows:

$$K = \frac{\sum_{i=1}^{N} C_i}{124 * N} \tag{2}$$

III. Experimental design and results

A. Experimental design

Different users have different consumption habits, hence, building a unified model for all users cannot perform effectively for all users. In fact, this has been confirmed in our original experiments. Consequently, considering the similarities among some user consumption behaviors, we use K-means to cluster users and divide user groups, and then we build model for each user group to predict their consumptions.

K-means is a typical cluster algorithm based on distance. We use K-means to cluster 50,000 users into 3 different clusters. The 3 clusters are as follows: Group0 has 29228 users, Group1 has 17154 users and Group2 has 3618 users. Their data rich degrees are as follows: Group0:10.95%, Group1:28.49%, Group2:46.59%.

We implement experiment for each user group, using former 123 weeks data to predict whether user consume in the 124th week, if consumes then recording as '1', if not, recording as '0'. We divide the data into training set and testing set as 9:1. The number of users in every user group's testing set and the number of users in each cluster are shown in table I. We mainly analyze the following 3 problems through our experiments: What is the effect of each algorithm on the prediction of time series when the data is sparse and has low data richness? What is the effect of adding the fusion features on the prediction results of each algorithm? What is the effect of the fusion between algorithms and how well they predict?

TABLE I
THE NUMBER OF USERS IN EACH CLUSTER

User group	Group0	Group1	Group2
0	1754	942	126
1	471	712	192
total	2228	1654	318

B. Experimental results and analysis

1) Experiment 1: Taking the consumption marks as features, we use SVM, LSTM and RNN respectively to train a model and predict for each user group. The prediction precisions are as follows:

TABLE II
EXPERIMENT 1 RESULTS

groups	Group0			Group1			Group2		
	SVM	LSTM	RNN	SVM	LSTM	RNN	SVM	LSTM	RNN
0	0.78	0.79	0.81	0.62	0.62	0.63	0.80	0.57	0.48
1	0.00	0.86	0.46	0.58	0.51	0.52	0.65	0.67	0.69
total	0.62	0.80	0.73	0.60	0.57	0.58	0.71	0.63	0.61

Group0's user data is sparse and very imbalanced. We obtain high precision in the 0 category with these algorithms, but SVM cannot predict the 1 category which is less proportion. LSTM can predict the 1 category with higher precision, which shows that LSTM has prominent effect on the unbalanced time series. RNN also can predict the 1 category, but RNN cannot filter memory information. Therefore, prediction ability of RNN is not so good as that of LSTM.

The data richness of Group1 and Group2 are increased, therefore, these algorithms can predict both of two types' users. From the result, by increasing the richness of the data can improve performance, especially in type which low proportion. Besides, RNN's advantageous benefit from the data's richness, so it's precision is slightly better than LSTM. From another angle, the Group2's data is rich than the Group1 so the prediction of Group2 is improved. However, we see that the data richness exceed certain degree will result in RNN slightly inferior to LSTM. The reason of this is that LSTM use the 'gate' to filter data which is help for remember the features, and eliminate redundancy. Experiment 1 proves the advantage of LSTM in memorizing long time series, when the data is sparse, unbalanced sequence. And RNN is not as good as LSTM due to long-term dependence. Besides, when data is abundant, SVM is more effective than LSTM and RNN. Then we try adding new features, and examine the effect of multi-feature fusion on each algorithm.

2) Experiment 2: We add the frequency features, using both of the consumption frequency and the consumption mark as the features in this experiment. The prediction results of each user group are as follows:

TABLE III
EXPERIMENT 2 RESULTS

groups	Group0			Group1			Group2		
	SVM	LSTM	RNN	SVM	LSTM	RNN	SVM	LSTM	RNN
0	0.79	0.79	0.80	0.61	0.62	0.57	0.71	0.57	0.00
1	0.67	0.86	0.60	0.59	0.51	0.65	0.64	0.67	0.60
total	0.76	0.80	0.76	0.60	0.57	0.60	0.67	0.63	0.36

After adding the frequency features, the effective features is further enriched. In Group0's results, SVM can predict the class 1 and achieve 67% precision. Adding features can improve SVM's prediction on the data which is sparse and the class with less proportion significantly. LSTM's prediction is better than SVM and RNN. For Group1, it can predict both classes, and maintain a stable forecast results. But the predictions of LSTM have declined. The features added may have an impact on LSTM's memory ability, and there is a certain amount of useless noise. RNN cannot filter the information, the noise doesn't have a great impact, and the prediction results is better than that of LSTM. SVM maintains a high prediction precision while the data is abundant. SVM in Group2 is same as in Group1, it shows a prediction advantage when the data is abundant, and achieves the highest prediction precision in the three algorithms. LSTM's precision is slightly lower than SVM. In other words, when the data is more abundant, LSTM's prediction results can be poorer than that of SVM. RNN's prediction result is abnormal, the prediction precision of class 0 is 0, which may be caused by overfitting. In our experiment, the neural network has 128 hidden units, which may be too many for the data. Hence, we try to reduce the number of hidden units to 64 and 32. Each precision of 64-RNN is: Class0:0.55, Class1:0.69 and total:0.64. And each precision of 32-RNN is: Class0:0.55, Class1:0.66 and total:0.62.

As we can see that RNN with 128 hidden units in the Group2 has the problem of overfitting. After adding the frequency features, the data richness increases to a certain extent, which has a negative impact on RNN. It appears overfitting faster than LSTM. In the same data size, LSTM can avoid the problem of overfitting very well through its own selective memory.

Comparing the results of experiment 1 and experiment 2, we can see that the prediction precision of LSTM on the three user groups does not change, and adding the frequency features has little effect on LSTM. Adding features and enriching the valid data can improve SVM's predictions on sparse and unbalanced time series. But when the data is too rich, excessive noise can adversely affect the prediction results of SVM. When the data richness is low, adding features can also improve RNN's result, but when the data is abundant, it is possible to make RNN appear overfitting if we continue to add features.

3) Experiment 3: On the basis of experiment 2, we continue to add the consumption amount as a feature, and implement experiments for each user group with consumption amount, consumption frequency and consumption mark as features. The prediction results for each user group are as follows:

TABLE IV
THE NUMBER OF USERS IN EACH CLUSTER

groups	Group0			Group1			Group2		
	SVM	LSTM	RNN	SVM	LSTM	RNN	SVM	LSTM	RNN
0	0.80	0.80	0.80	0.57	0.62	0.63	0.00	0.56	0.50
1	0.92	0.79	0.63	0.00	0.52	0.50	0.60	0.68	0.65
total	0.82	0.79	0.76	0.32	0.57	0.58	0.36	0.63	0.59

After adding consumption amount, features are increased and the data for training and testing is further enriched. The three algorithms all can predict well on Group0, and SVM still maintains its advantages when the data is abundant. Likewise, LSTM is superior to RNN in terms of long-term memory. SVM's precision on Group1 has dropped significantly, it cannot predict the less proportion class 1. According to the mutations of RNN in experiment 2, we have a reason to think that SVM has been overfitted to cause the exception. Both LSTM and RNN can predict the consumption behaviors, and similarly as in the experiment 2, RNN's precision is slightly higher than that of LSTM. SVM also appears abnormal situation on Group2 as same as on Group1. The precision of LSTM is higher than RNN on Group2. LSTM has advantages over long time series, and its filtering memory capabilities for abundant time series play an effective role.

Comparing the results of the above three experiments, we can see that LSTM's prediction precision decreases by 0.01 only in Group0, while remains stable for all the others, which further proves that adding features has little effect on LSTM. As mentioned before, the three features used in the experiments have some redundancy. For instance, if a user has consumption mark of 1, then he certainly has consumption frequency and consumption amount features. There are no feature data when users don't consume. This redundancy may be the reason why multi-features cannot work for LSTM. In experiment 3, it can be seen that SVM's prediction effect can be improved after adding some detailed features. However, when features are added to a certain extent, the redundancy will adversely affect SVM, which will cause SVM overfitted. RNN overfits in experiment 2, but in experiment 3, after adding consumption amount features, it doesn't overfit. Taking into account that the absolute range of consumption amount is large, we normalize the amount, scale it to the range of [0, 1]. Experiment is implemented for Group2 again in order to verify its impact, and the results are: Class0:0.40, Class1:0.00 and Class:0.16.

After normalization, we can see that RNN has been overfitted more seriously, and its prediction is further deteriorated. Therefore, we believe that the absolute range of the consumption amount is much larger compared with that of consumption mark and frequency, which has an influence on RNN's fitting, making RNN does not appear overfitting in experiment 3.

In conclusion, we can see that the addition of features can effectively improve the performance on sparse time series and the imbalanced category, especially for the performance of SVM. However, when the data is abundant to a certain extent, the redundant features will adversely affect the prediction of SVM and RNN, their prediction precisions can be reduced, and SVM and RNN will be overfitted. The added features have little effect on LSTM's prediction, therefore, through the effective information filtering, LSTM can also avoid the problem of overfitting.

4) Experiment 4: In the above experiments, we only use single algorithm to predict consumptions. We can see that each algorithm has its own advantages and disadvantages. Then we consider to fuse LSTM and SVM, and inspect whether it can achieve the purpose of learning from each other. We consider using LSTM to extract the feature of the time series then use its classification probability as the new features input into SVM to predict. We refer to this fusion algorithm as LSTM-SVM. Similarly,we exchange the LSTM and SVM in LSTM-SVM then refer as SVM-LSTM. We use these two fusion algorithms to re-implement the experiments 1 to 3, and the results are as follows:

TABLE V
EXPERIMENT 1 RESULTS OF FUSION ALGORITHMS

groups	Group0		Group1		Group2	
	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM
0	0.79	0.81	0.59	0.63	0.82	0.71
1	0.89	0.43	0.56	0.55	0.63	0.65
total	0.81	0.73	0.58	0.60	0.71	0.67

As we can see, both fusion algorithms can predict both categories. For Group0's average prediction precision, we have LSTM-SVM>LSTM>RNN=SVM-LSTM>SVM. Since LSTM can play its own memory ability when extracting features, and SVM can also play its own advantages on classification, hence the LSTM-SVM can achieve the best predictive results. As to SVM-LSTM, it acquired a part of the memory capability of LSTM. Thus it can predict the class 1, and greatly improve SVM's predictions on the less proportion categories.

For Group1's average prediction precision, we have SVM=SVM-LSTM>LSTM-SVM>RNN>LSTM.
SVM maintains the advantages when data is abundant. For Group2's average prediction precision, we have LSTM-SVM=SVM>SVM-LSTM>LSTM>RNN. At this time both SVM and LSTM-SVM have the same prediction precision. Through the integration with SVM, LSTM can also have SVM's classification advantages on the abundant data.

In general, for the sparse data, LSTM-SVM fusion method enhances the forecast significantly, while for the abundant data, SVM has a better classification advantages.

TABLE VI
EXPERIMENT 2 RESULTS OF FUSION ALGORITHMS

groups	Group0		Group1		Group2	
	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM
0	0.79	0.80	0.59	0.63	0.71	0.51
1	0.83	0.62	0.61	0.54	0.63	0.65
total	0.80	0.76	0.60	0.59	0.66	0.59

For Group0's average prediction precision, we have LSTM-SVM=LSTM>SVM=SVM-LSTM=RNN. LSTM-SVM still maintains the LSTM's memory and SVM's classification advantages, thus getting the highest prediction precision. For Group1's average prediction precision, we have LSTM-SVM=SVM=RNN>SVM-LSTM>LSTM. For Group2s average prediction precision, we have SVM>LSTM-SVM>LSTM>SVM-LSTM>RNN.
It can be seen that after the data becomes more abundant, SVM still has a good prediction results, LSTM-SVM in Group1 is as same as SVM, and in Group2 its precision is only 0.01% lower than that of SVM.

TABLE VII
EXPERIMENT 3 RESULTS OF FUSION ALGORITHMS

groups	Group0		Group1		Group2	
	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM	LSTM-SVM	SVM-LSTM
0	0.79	0.80	0.59	0.00	0.77	0.40
1	0.83	0.85	0.59	0.43	0.63	0.00
total	0.80	0.81	0.59	0.19	0.69	0.16

For Group0's average prediction precision, we have SVM>SVM-LSTM>LSTM-SVM>LSTM>RNN. SVM still has its advantages with the highest precision when adding features and the effective data increased. In Group1, SVM has been overfitted, which is the same as SVM-LSTM. The difference is that SVM is unable to predict the less proportion category, while SVM-LSTM is unable to predict the larger proportion category. This may be due to after LSTM filtering the information, the remain memory information is different from SVM. For Group1's average prediction precision, we have LSTM-SVM>RNN>LSTM>SVM>SVM-LSTM. LSTM-SVM effectively avoids overfitting. In Group2, SVM and SVM-LSTM are overfitted as same as in Group1. For Group2's average prediction precision, we have LSTM-SVM>LSTM>RNN>SVM>SVM-LSTM. LSTM-SVM still has the highest precision.

In the above three re-implemented experiments, there are 9 prediction results for the three user groups. LSTM-SVM has the highest precision of 6 results and SVM has the highest precision of 5 ones, and they have the same precision twice. We can see that both LSTM-SVM and SVM algorithms have the best forecast results for most cases. SVM has very good predictions when the data is abundant, but it is often not predictable for sparse time series, and SVM will appear overfitting problem when the data is abundant in some case. Overall, LSTM-SVM has significant advantages over other algorithms with sparse data and unbalanced time series. Moreover, after the data is abundant, it can also effectively predict the results. There are no obvious shortcomings in LSTM-SVM, and it can effectively avoid overfitting problems.

IV. CONCLUSION

In this paper, we use real user consumption records to construct time series, and exact variety features to combine with machine learning algorithms to achieve the consumption prediction. We consider the impact of different features on various algorithms and then achieve multi-feature fusion and algorithm fusion to complement the weakness of each single feature and algorithm. This study provides the basis for further study. At the same time, it necessary to consider the time series various characteristics and application of predictive model in the future.

ACKNOWLEDGMENT

This work was supported by the National Social Science Foundation of China under Grant 16ZDA055; National Natural Science Foundation of China under Grant 91546121, 71231002 and 61202247; EU FP7 IRSES MobileCloud Project 612212; the 111 Project of China under Grant B08004; Engineering Research Center of Information Networks, Ministry of Education; the project of Beijing Institute of Science and Technology Information; the project of CapInfo Company Limited.

REFERENCES

[1] L. Qi, *Time Series Analysis Simple Course*. Tsinghua University Press, 1999.
[2] Y. Wang and L. Wang, *Application Time Series Analysis*. China Renmin University Press, 2005.
[3] X. Xu, "Research and application on time series prediction based on data mining method." Ph.D. dissertation, China University of Geosciences for Master Degree(Beijing), 2011.
[4] T. Le, Y. Cai, X. Ma, and L. Wang, "Development and application of time series forecasting," *Ordnance Industry Automation*, no. 2, pp. 63–68, 2015.
[5] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, and L. Buydens, "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35–49, 2003.
[6] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. IEEE, 2015, pp. 153–158.
[7] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *Chinese Association of Automation (YAC), Youth Academic Annual Conference of*. IEEE, 2016, pp. 324–328.
[8] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake prediction based on spatio-temporal data mining: An lstm network approach," *IEEE Transactions on Emerging Topics in Computing*, 2017.
[9] J. Fan, Q. Li, Y. Zhu, j. Hou, and y. Feng, "A spatiotemporal prediction framework for air pollution based on deep rnn," *Science of Surveying and Mapping*, pp. 1–16, 2017(07).
[10] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.
[11] H. Li, *Statistical Learning Method*. Tsinghua University Press, 2012.
[12] A. Cimino and F. Dell'Orletta, "Tandem lstm-svm approach for sentiment analysis." in *CLiC-it/EVALITA*, 2016.
[13] Z. Zhou, *Machine Learning*. Tsinghua University Press, 2015.