International Conference on Recent Innovations in Science, Engineering & Management
R.R. INSTITUTE OF TECHNOLOGY, Bengaluru (India)
10th - 11th May 2019 www.conferenceworld.in ISBN : 978-93-87793-87-3

Reconstruction of Lost Data over Uneven Time Intervals Using Apache Spark

Kishan S. Divate[1], Shreya V. Puthi[2], Pallavi Gowdoor[3]

U.G. Student, Department of ISE, SKIT Engineering College, Bangalore, India[1]

U.G. Student, Department of ISE, SKIT Engineering College, Bangalore, India[2]

Department of ISE, SKIT Engineering College, Bangalore, India[3]

ABSTRACT

A larger volume of data has been generated in many areas than in the past. However, the lack of some values in the collected data always occurs in practice and calls into question the extraction of the maximum value of these large-scale data sets. However, in uneven time intervals, most existing methods could be unachievable or could be inefficient in predicting lost data. In this paper, we have taken the challenge of reconstructing the lost data in uneven time intervals using improved matrix factorization techniques. Our approaches are optimally designed to use to a large extent the internal patterns of each time intervals and information on time intervals in multiple sources. Based on the idea, we imposed three different regularization terms to restrict the objective functions of matrix factorization. Extensive experiments on real data sets and synthetic datasets show that the proposed approaches can effectively improve the performance of reconstruction of lost data in uneven time intervals. Furthermore, we also demonstrated how to leverage Apache Spark's high processing power to make reconstruction about lost data in large-scale uneven time intervals.

Key words: Apache Spark, Big Data, matrix factorization, Reconstruction of Lost Data, Uneven Time Intervals.

I. INTRODUCTION

The sophistication of data collection tools from multiple sources and the resulting volume of data has grown to an unprecedented level since the era of big data began in recent years. Uneven time intervals, a common data format, are ubiquitous in many real-world applications, such as monitoring electrical equipment, weather or economic forecasting, monitoring the state of the environment, security surveillance and many others. In most applications, multiple sensors are used to generate time intervals data and generally share a common goal. For example, in the electrical network system, various diagnostic gas sensors are implemented to monitor the status of the main power transformers and generate uneven time intervals by measuring the content of diagnostic gases over time. In the Internet of Things, many sensors are used to produce uneven time intervals from the external environment, such as air or water quality. In medical and health care systems, more sensors can be equipped within residential spaces to monitor the health and general well-being of older people, while ensuring adequate

care and assistance to people who recover lost mobility also through therapy. In this paper sensors that share a common goal are treated as a network of sensors.

Unfortunately, due to the harsh working conditions or uncontrollable factors such as extreme weather, breakdowns or unstable communication signal, the time intervals crude sensor network usually involves lost values. For example, during service, missing values in network monitoring systems can occur for various reasons, such as rapid acetylene evaporation, the existence of contamination on the surface of the platinum alloy of a gas meter, etc. Sensors and communication failures are more common factors that produce missing values in many applications. And what is worse, the immediate resolution of these practical problems is rarely plausible and can cost too much. The inevitable lack of data requires an integrated analysis of the observed data sets. A large collection of statistical methods and data extraction has been proposed to predict missing time intervals values.

In this paper, with the aim of solving the above problems, we propose to combine the temporal softness of time intervals and information through multiple sources in matrix factorization to improve the accuracy of reconstruction of lost data in uneven time intervals. First, since each time intervals rarely fluctuates violently over time, i.e. time intervals generally contain an internal and tangible model of temporary softness. Therefore, we try to exploit the feature to reduce the reconstructing error of the lost data in the uneven time intervals. Specifically, in the objective function of matrix factorization, a selective penalty term is used to standardize the time intervals, i.e. our objective is to minimize the fluctuation of each time intervals over time. Secondly, since valuable correlation information exists on multiple sources in a sensor network, we also try to combine this information into matrix factorization for better performance. Specifically, the correlation information is incorporated in the design of two terms of regularization of the sensor network, i.e. the regularization term based on correlated sensors and the regularization term based on unrelated sensors, for Limiting the objective function of matrix factorization. Wet to minimize the difference between the sensor and its related sensors or to maximize the difference between a sensor and its unrelated sensors depending on the regularization of the sensor network. Furthermore, to treat related or unrelated sensors differently, we have further improved the regularization terms of the sensor network of the objective function that incorporates similarity functions.

In the era of big data, more massive volumes of time intervals data are being generated than ever. Furthermore, the analysis of big data is a complex and slow activity, which requires a more efficient and specific analysis tool than the traditional ones. Therefore, the parallel versions of matrix factoring have become of great interest. Apache Spark is a large-scale distributed data processing environment based on the principles of scalability and fault tolerance that are fundamental to the success of Hadoop and MapReduce apache Spark has already implemented a fundamental version of matrix factorization for recommendations. Here, we implement our proposed approaches using the Apache Spark platform. The experimental results reveal that our proposed methods show superior performance compared to traditional algorithms and the state of the heart. The contributions of this document are summarized below: 1) We propose new methods to limit the factoring of the matrix by combining the temporal fluency of each time intervals and the information coming from multiple sources to improve the forecasting performance of lost data in uneven time intervals. These restrictions are used

to significantly exploit the internal characteristics of time intervals data. 2) We develop how softness constraints are carefully designed and how correlation information between different sources in a sensor network can contribute to the reconstruction of lost data in uneven time intervals. Furthermore, we have incorporated softness restrictions and two terms of regularization of the sensor network to limit matrix factorization, respectively. Furthermore, we illustrate systematically how to design matrix objective factorization functions with carefully designed regularization terms. 3) We implement and verify the proposed methods with three sets of real data and an interval of synthetic data. And for big data analysis, we also implement and verify the methods proposed in the Apache Spark platform.

II. PROPOSED WORK

In this section, we describe the details of the proposed methods for predicting lost data in uneven time intervals. Let's start with the discussion on the reference solution for the problem. Subsequently, we have elaborated the key idea of the proposed methods. We present how to take advantage of the softness feature to reduce the error of reconstruction of lost data in uneven time intervals. Furthermore, we also explain why and how to use valuable correlation information in multiple sources in time intervals data collected by a network of sensors to improve forecasting performance. Given the main idea, we carefully designed three regularization terms to limit matrix factoring and then build five different models. In the process of designing regularization terms, five similarity functions are introduced, which are also key components of the proposed method. Finally, we give the detailed implementation of the proposed methods based on Apache Spark.

2.1 Architecture of proposed model

The architecture of our prototype is shown in Fig-2.1.The proposed method can be applied to any network fields. It can reconstruct the lost data of any fields like, online cab booking (ola, uber), military, hospitals etc. The data can be of any formats and can be applied dynamically. It predicts the lost data based on the input factors.

Initially the raw datasets are fetched from source. These datasets are stored in Hadoop distributed file system for future processing purpose. The datasets can be in any format like example, structured, unstructured, or semi structured way. These datasets have to be formatted into one format. The datasets are then forwarded to spark for reconstruction lost data over uneven time intervals. The use of hive in spark makes it better for accessing the datasets and processing it. Spark is an efficient tool for processing bigdata. Then we apply the algorithm for prediction of lost data. The reconstructed data is then stored in hdfs for future purposes.

2.2Low rank matrix factorization

Singular values (SVD) is a popular and effective in a real or complex matrix factorization. The SVD approach focuses on the discovery of linear correlations between time intervals and the application of these correlations for further data analysis. orthogonal columns are a diagonal matrix which contains the singular values along with its main diagonal. The most popular low rank factorization is achieved when SVD is reorganized.

2.3 Fusion of the temporal smoothness of the time intervals

In the real world, a large number of time intervals usually do not fluctuate in an uncontrolled way. For example, ambient temperature, gas concentrations in electrical equipment, energy consumption in cities and prices of products on the market rarely change drastically. To combine the temporal smoothness of each time intervals, since V indicates the latent matrix with the time dimension, the optimization problem is improved. Since Alternate Least Squares (ALS) can be done effectively, the key idea is to find the optimal local solution.

2.4Fusion of information across multiple sources

Several sources provide access to information about raw data nearby. First, we strive to merge valuable information into multiple sources by integrating related sources. So, from the opposite point of view, unrelated sources also provide us with a meaningful view of the distant raw data.

2.4.1 Regularization based on related sensors

In a sensor network, although the different sensors are assigned to different tasks, they generally share a common goal and there may be a strong correlation between some sensors. For example, in environmental monitoring systems, there may also be a high correlation between chemical and biological sensors, as their detection values can change simultaneously. In terms of personal medical care, blood pressure generally increases with heart rate, so the corresponding sensors can have a strong correlation. If one sensor has a strong correlation with another, we call that the two sensors are correlated. The proposed model for predicting lost data can optimize the problem.

2.4.2 Regularization based on unrelated sensors

The proposed model imposes a regularization term based on correlated sensors to limit matrix factoring. From the opposite point of view, if one sensor has a weak correlation with another, we call the two sensors that are not correlated. Furthermore, we use another term for the regularization of the sensor network, ie the regularization term based on unrelated sensors, to build the matrix factorization model based on unrelated sensors. Since unrelated sensors share a weak correlation, we try to add a regularization term to maximize the distance between the sensor and its unrelated sensors.

2.4.3 Integration of temporal smoothness of time intervals and information across multiple sources

The previously proposed CSM, USM and MFS models aim to exploit information through multiple sources or the temporal fluency of time intervals. Of course, it is convincing that the combined fusion of the two characteristics of uneven time intervals can also contribute to improving the performance of lost data reconstruction. But the proposed model can overcome the disadvantage of the existing models. The proposed model can achieve the prediction of lost data of multiple sources at uneven time intervals.

2.4.4 Implementation of the methods proposed in Apache Spark

The scale of modern time intervals data sets is growing rapidly. And there is an urgent need to develop solutions to exploit this large amount of data using statistical methods. Spark is a distributed computing framework developed at UC Berkeley AMP Lab. The parallel execution model in the Spark memory in which all the data will be loaded into memory to prevent the I/O bottleneck from benefiting from the iterative calculation. Spark also provides very flexible data flows based on DAG (direct acyclic graph), which can

significantly accelerate the calculation of iterative algorithms. The two features of Spark lead performance up to 100 times faster than the Hadoop MapReduce paradigm in two phases. Here, we implement our proposed methods on the Apache Spark platform. To make the solutions more adaptable to the platform, the gradients of the objective functions are rewritten in matrix form.

2.4.5 Overall Algorithm

Algorithm: Generic method for reconstruction of lost data over uneven time intervals.

Input: uneven time series A, indicator matrix B

(parameters: S, V, L, λ, β, and γ)

Output: reconstructed data.

1. initialize data frames A and B as RDD
2. Data filtration (using spark methods)
3. Data Aggregation (using spark methods)
4. repeat
5. if X= corelated data go to step 6 else go to 10
6: γ = computing the best step size;
7: for i = 1 to N do
8: Si = Si - γ (∂LV/ ∂Si) ▷ based on MFS Equation
9: end for go to step 18
10.rddA,rddB ← SparkContext.textFile(dataPath)
11: rowMatrixA ← new RowMatrix(rddA)
12: calculate (∂LV/∂S) and (∂LV/∂V) ▷ based on USMS Equation
13: S = S.map {Si - γ ∂L′SV ∂Si} ▷ updating S
14: V = V - γ ∂L′SV ∂V = V - γ ∂LV ∂V ▷ updating V
15: until Convergence
16: rowMatrixS ← new RowMatrix(rddS)
17: X = rowMatrixS.multiply(V T).collected()
18: reconstructed data
19: end if

The generic algorithm to solve the problem illustrated. As the Algorithm shows, given the uneven time intervals A, the indicator matrix B, the algorithm is designed to obtain a more accurate solution. Initially the raw data is filtered according to the needs using spark filter method. Then aggregation is applied. If the dataset is a corelated then step 6 is performed else step 10. If the data is uncorrelated then algorithm updates γ until convergence and the step size is updated in each iteration based on the search strategy of the line. The lost

values could be obtained from the planned A. The A and B input data are then transformed into a robust distributed data set (RDD), ierddA and rddB, respectively, which is a new distributed memory abstraction in Spark. Therefore, to implement the multiplication of the matrix in Spark, rddA is transformed as Row Matrix so that it can be multiplied by a local matrix. Subsequently, Si and V are updated from the gradients calculated up to convergence. Finally, the expected X is obtained by multiplying Row Matrix once more.

III. FIGURES

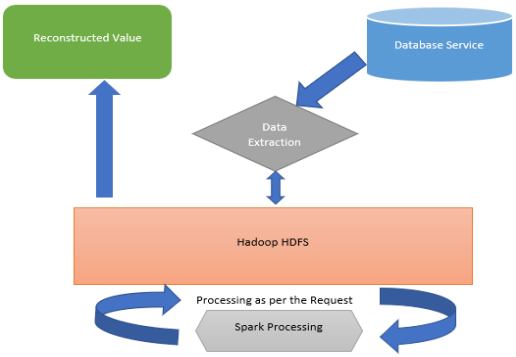

Fig-2.1: Proposed Architecture

IV. CONCLUSION

In this paper, we have proposed new method for limiting matrix factorization to predict lost data in time intervals from multiple sources, which results in satisfactory provision of lost data reconstruction and high calculation efficiency. The methods aim to merge the softness characteristics of each time intervals and the valuable correlation information through multiple sources in a network of sensors in matrix factorization. Correspondingly, the methods incorporate softness restrictions, to optimize the matrix factoring solution. The evident superiority of the proposed model reveals the effectiveness of the extraction of latent factors in the matrix factoring process after having incorporated the constraints. Furthermore, the combination of information extraction in multiple sources and the temporal fluency of each time intervals demonstrate the effectiveness of the proposed methods. Even when the missing proportion is 90%, the RMSE of the proposed methods is within a reasonable range. We conclude that the proposed methods are alternative models for predicting missing values in large-scale uneven time intervals.

V. ACKNOWLEDGEMENT

With all respect and gratitude, we would like to thank all the people who have helped us directly or indirectly for the completion of the paper " Reconstruction of Lost Data Over Uneven Time Intervals Using Apache Spark". We express our heartily gratitude towards Prof. Pallavi Gowdoor for guiding us to understand the work conceptually and also for her constant encouragement to complete the paper. Our association with her as a student has been extremely inspiring. We would like to give our sincere thanks to Dr. Hemalatha K.L. Head of the Department of Information Science and Engineering for her technical support and constant encouragement.

We would also like to extend our sincere thanks to our Principal Dr. Manjunatha A. for his help and support in all respects. We would also like to thank all our staff members and collogues who helped us directly or indirectly throughout our dissertation work.

REFERENCES

[1] H. Nguyen, W. Liu, F. Chen, "Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data", IEEE Transactions on Big DataPP (99) (2016) 1–1.

[2] Y. Cai, H. Tong, W. Fan, P. Ji, "Fast Mining of a Network of Coevolving Time Series", in: Proceedings of the 2015SIAMInternationalConference on Data Mining, pp. 298– 306.doi:10.1137/1.97816119 74010.34.

[3] P. Baraldi, F. D. Maio, D. Genini, E. Zio, "Reconstruction of Missing Data in Multidimensional Time Series by Fuzzy Similarity", in: Proceedings of the SIAM International Conference, 2016.

[4] N. Meger, C. Rigotti, C. Pothier, "Swap Randomization Of Bases Of ´Sequences For Mining Satellite Image Times Series", in: Proceedings of theEuropean Conference on Machine Learning and Knowledge Discoveryin Databases (ECML PKDD), Springer, 2015, pp. 190–205.

[5] J. C. Jacob, D. S. Katz, G. B. Berriman, J. C. Good, A. C. Laity, E. Deelman, C. Kesselman, G. Singh, M.-H. Su, T. A. Prince, and R. Williams, ―"Montage: A Grid Portal And Software Toolkit for Science-Grade Astronomical Image Mosaicking",‖ International Journal of Computational Science and Engineering, vol. 4, no. 2, pp. 73–87, 2009.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, ―Basic local alignment search tool, ‖ Journal of Molecular Biology, vol. 215, no. 3, pp. 403–410, 1990.

[7] R. M. Bell and Y. Koren, ―Lessons from the Netflix prize challenge, ‖ SIGKDD Explor. Newsl., vol. 9, no. 2, pp. 75–79, 2007.

[8] C. C. Johnson, ―Logistic matrix factorization for implicit feedback data, ‖ in NIPS Workshop on Distributed Machine Learning and Matrix Computations, 2014.

[9] M. Deshpande and G. Karypis. Item-based top-n recommendation. ACM Transactions on Information Systems, 22(1):143–177, 2004.