Application of Semantic Web Technologies for Learning Pattern of Storm Damages

Quang-Khai Tran, Sa-Kwang Song
University of Science and Technology (UST),
Daejeon, Republic of Korea
Korea Institute of Science and Technology Information
(KISTI), Daejeon, Republic of Korea

Jung-Ho Um
Korea Institute of Science and Technology Information
(KISTI)
Daejeon, Republic of Korea

***Abstract*— It seems that Semantic Web technologies have not had strong connections with problems of natural disaster prediction, such as storm forecast, which remains a big challenge for science. Aiming to estimate potential damages of storms to societies and economic systems in the view of Machine Learning and Data Mining, we suggest an investigation of storm data in the SRBench dataset using Statistical Unit Node Set approach, to learn five hurricanes in The USA. Our study considers some locations affected by the hurricanes to construct data matrices, in which the rows indicates time instances of storm-situation, while the columns represent weather observations at locations, attributes of storms and known and unknown damage indices of each location. Two matrix completion algorithms are used to perform stream-reasoning for filling the unknown entries, which can be considered as predicted values of potential losses. One of the biggest obstacles is the availability and quality of data, which are incomplete and carry noises. However, results show that our model can deal with this issue and reflect the pattern of storm damages. Therefore, we recommend this achievement as a meaningful alterna-tive to, or support to, meteorological models for governmental and business organizations, especially in developing countries where infrastructure and resources of weather forecasting are limited.**

Keywords*— Matrix completion; statistical unit node set; storm damages prediction.

I. INTRODUCTION

One of our world's biggest issues today is the increasing situation of Global Climate Change, leading to more and more severe and dangerous natural disasters including storm, flood and drought. Particularly, many countries and regions have experienced huge damages in recent years because of storms. For example, hurricane Katrina (in The USA in 2005), which is considered as one of the most deadly storm in the USA history, caused more than 1,800 deaths and $108 billion of economic loss [1]. Elsewhere, similar or even more destroying storm disasters happen seemingly every year, like super typhoon Haiyan (Philippines, 2013) and cyclone Nagis (Myanmar, 2008) both caused many thousands of deaths and missing people [2]. Therefore, prediction of such natural disasters is undoubtedly important to reduce damages, particularly in term of human lives and socio-economic values. However, storm forecasting, especially of intensity and track, still remains a challenge for science [3], even with modern meteorological numerical models which are supported by super computing systems and various kinds of data. For the North Atlantic Ocean area, mean error of forecasting the track of tropical storms is more than 250 sea-miles (for 120 hours interval) [4]. But the real inaccuracy of prediction can be higher depending on seasons and geographical regions, as well as temporal and spatial scales (the longer time and/or the smaller region, the more difficult for forecasting).

Aiming to contribute to tackling this problem, our study suggests an application of Semantic Web (SW) technologies for predicting storm damages in the manner of Machine Learn-ing and Data Mining. This paper presents a continuation of our previous work [5], in which we stopped at preliminary tests on very limited data of only hurricane Katrina. In this work, we consider data of five severe hurricanes in The USA provided in the SRBench data set [6], and suggest the way how storm data can be trans-coded into matrices for effective computation, as well as exercise tests on some configurations. In more details, our idea is to investigate how much a coming storm can affect some locations of interest (e.g.: counties in The USA) by a regression on weather data in those locations and the storm's observation data. Fig. 1 shows an example of the path and variations of hurricane Katrina in terms of direction, power (the bigger the circles are, the more powerful the hurricane is) and affected regions. Our model is not to compute the movement and fluctuating power of hurricanes, but to focus on locations. We believe that this predicting strategy is practically effective for forecasting effects of future storms by SW resources.

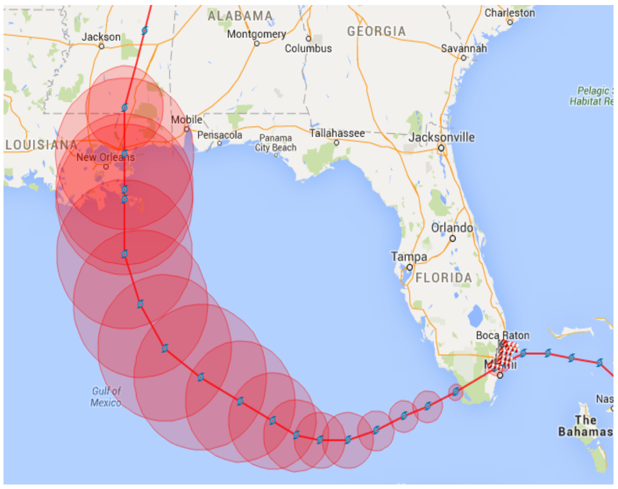


Fig. 1. A part of the track of hurricane Katrina.

In term of methodology, we employ Statistical Unit Node Set (SUNS) framework (introduced in [7]) to trans-code data in triple form (*subject*, *predicate*, *object*) into matrices, and use its Matrix Completion (MC) methods for filling missing entries in the matrices to imply the predicted values of Storm Damage Index (SDI). SDI value of each location is calculated based-on its population size, and the hurricane's power and damage degree. In next section, more detailed about SUNS approach and SDI calculation will be described.

II. THEORETICAL BACKGROUND

A. *Statistical Unit Node Set Framework*

SUNS approach is a comprehensive framework for perfor-ming statistical relational learning with SW data, and applied in [8] and [9] (the same research group with [7]) for investigating *human-to-human* and *gene-disease* relationships in Resource Description Framework (RDF) data. In SUNS, entities (such as people or genes and diseases) which have attributes and relate to variables of interest are called "statistical units", and a set of samples of statistical units is a "population" ("node" indicates entities or their relationships in ontology graph). The authors utilized SPARQL query to retrieve necessary information to form data matrices and MC models to perform multivariate regression. The framework has successfully shown its ability in learning sparse data matrices to exploit hidden relationships in *friend-of-a-friend* (FOAF) ontology, YAGO2 ontology (to predict nationality of writers in data collected from Wikipedia, WordNet and GeoNames), Linked Life Data and Bio2RDF in the Linked Open Data resources [9].

One of the most common MC algorithms is based-on Singular Value Decomposition (SVD), as given below:

$$X = U_r D_r V_r^T \tag{1}$$

In which X is a mxn data matrix with reduced-rank r (and $r < min(m,n)$), U_r (mxr) and V_r (nxr) are orthonormal matrices formed from eigenvectors of XX^T and X^TX respectively, and D_r (rxr) is a diagonal matrix formed from the r-biggest eigenvalues. Missing entries in X are filled by a model matrix Y (of mxn dimension), which is considered as a generalization of X, reconstructed from the three low-rank factorization components in (1).

In [8], authors introduced a new algorithm named Reduced-Rank Penalized Regression (RRPR, but they used the abbrevia-tion name "RRPP") which can perform better regression:

$$\hat{Y} = U_r \operatorname{diag}_r\left(\frac{d_k}{d_k + \lambda}\right) U_r^T Y \tag{2}$$

Where $\hat{Y}$ is an approximation of Y (and so X), U_r is the same with that in (1) and $\operatorname{diag}_r\left(\frac{d_k}{d_k+\lambda}\right)$ is derived from D_r in (1) (with d_k is the k*-th* eigenvalue and λ is a given small number lower than 1). They also tested two other MC methods called Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NNMF), and the comparison with SVD-based and RRPR algorithms indicated that RRPR provided the best prediction (refer to [8] for more details).

B. *Matrix Completion Algorithms*

We apply two MC algorithms, SVD-based and RRPR, with a purpose to compare a state-of-the-art method and a traditional one as the baseline. In [5], we adapted the two algorithms from *SVD-Impute* [10] and *SOFT-Impute* methods [11], and named the SVD-based approach as "Naive" SVD (given below):

```
Algorithm: "Naive" SVD
($\Omega$ denotes the set of non-zero entries in $X$)
    Step 0: set $\hat{Y} = 0$
    Step 1: $Y_{ij} = \begin{cases} X_{ij} \text{ if } (i,j) \in \Omega \\ \hat{Y}_{ij} \text{ otherwise} \end{cases}$
    Step 2: $U_r$, $D_r$ and $V_r$ are derived from SVD($Y$), with
            $r < min(n,m)$
    Step 3: $\hat{Y}$ is reconstructed by (1)
    Step 4: if $\hat{Y}$ is not converged, go to Step 1
```

RRPR can be obtained by simply replacing formula (1) by formula (2) in ***Step 3***. The convergence threshold for both algorithms is based-on relative error (RELERR) between non-zero entries in $\hat{Y}$ and X:

$$\text{RELERR}(\hat{Y}, X) = \sqrt{\frac{\sum_{(i,j)\in\Omega_X} (\hat{Y}_{ij} - X_{ij})^2}{\sum_{(i,j)\in\Omega_X} X_{ij}^2}} \tag{3}$$

The algorithm can stop when RELERR is lower than a small given ϵ ($0 < \epsilon < 1$). Another measure of error commonly used in MC-related studies is Root Mean Square Error (RMSE):

$$\text{RMSE}(\hat{Y}, X) = \sqrt{\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega_X} (\hat{Y}_{ij} - X_{ij})^2} \tag{4}$$

An usual issue of meteorological models is that they need good and sufficient data for making good prediction, but this requirement is not always satisfied. For example, we observe that when hurricane Katrina hit Florida, it destroyed most of the weather sensors and there was no observation right under its position. However, multivariate learning used in SUNS has been seen as a strong way to deal with incomplete data. This is critically important in our study, as SW resources may lack of a lot of information and contain noises.

C. *Storm Damage Index*

Economic and human losses that different locations have to suffer under a storm are not the same, but very various based on many conditions, such as socio-economic characteristics, infrastructures, population size and density, how far from the storm eye... An exact estimation of all kinds of losses seems to be impossible to make. Hence, we suggest a damage index (the above mentioned SDI) which can reflect different ranges of losses for separate locations, based-on different study purposes. So far, we aim to estimate the damages for local residence, and calculate SDI value by the following summation:

$$\frac{StormLevel + StormDamageLevel}{2} + PopulationRate \tag{5}$$

The storm level relates to the strength or grade of the storm, and can be referred to its wind-speed to map with given ranges. The storm damage level can be referred to different kinds of losses, or simply the general loss and distance to the storm center. The population rate is the residential density of the considered location. Details of these three parameters when being adapted into our model will be explained in section IV.

III. LEARNING AND PREDICTING MODEL

The SRBench dataset is constructed mainly by data from Linked Observation Data, Linked Sensor Data, DBPedia and GeoNames datasets of the Linked Open Data cloud [6, 12]. The hurricane data (from Linked Observation Data and Linked Sensor Data datasets) consist of triple-form information of 7 hurricanes and 1 blizzard in The USA. However, we just can query data of 5 hurricanes, including: Charley (2004), Katrina (2005), Wilma (2005), Gustav (2008) and Ike (2008), with totally 15,032,665 observation data lines. In addition, we see that most of sensor stations are located in 336 counties on or near the coastal line. As the data have the same RDF structure with tested data of SUNS framework, we employ its two main components, one is forming data matrices and then performing matrix completion, to build the learning and predicting model.

Fig. 2 presents the general structure of the model, in which we consider two essential concepts "*location*" and "*storm*" as statistical units, and their relationship is the target of the prediction procedure. With inputs from SRBench, "*county*" and "*hurricane*" are reflected as instances of the two concepts, respectively. Therefore, columns of the data matrix are dedica-ted to represent weather data at counties and observed data of hurricanes. And as a result, data matrices of separate hurricanes can be aggregated by vertically combining, rather than horizon-tally, to build the data matrix for the regression process.

To form the matrix's rows, we take advantage of the data in streaming form. At a concrete time, all information about the weather situation at locations and attributes of a storm form a row of the matrix (a time instance). Also at that time, relation-ship of "location" and "storm" is reflected by the damage that the storm (S) causes to the location (L), indicated by the triple (S, *causes-damage-to*, L) or the extended triple with time stamp (S, *causes-damage-to*, L, T_i) [5]. Value of the triple is assigned with the SDI value at time T_i (0 if there is no effect). However, an issue appears that time periods of this streaming data are not always consistent. Data of hurricane Ike is available at each 1-minute, but data of other 4 hurricanes is set at each 5-minute.

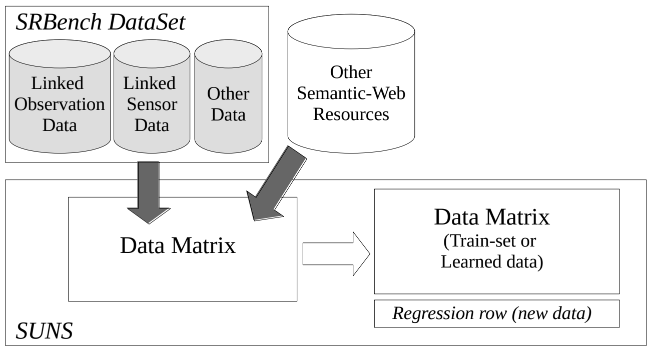


Fig. 2. Learning and predicting model.

IV. DATA MATRICES

A. *Matrix of Hurricanes' Observation Data*

Firstly, properties of the *subject* (the concept "storm" in this case) are reflected into data matrix. 9 attributes of hurricane are chosen to form 9 columns of the matrix, consisting of latitude, longitude, category (grade or degree or level of the storm), wind-speed, pressure, hurricane radius, hurricane ratio, tropical storm radius and tropical storm ratio. The last 4 attributes are implied from the fact that not all parts of a hurricane have the same intensity, and the destructive power is various depending on distance to the hurricane eye. Consequently, a row vector of 9 variables is to represent a time instance of the situation of the happening storm event. And m_k streaming data lines of a certain hurricane will form an m_kx9 matrix. Fig. 3a shows the aggregated *observation data matrix* of different data matrices (let us call it "the matrix **A**"), with data entries are written into the grid cells (0 if there is no corresponding information).

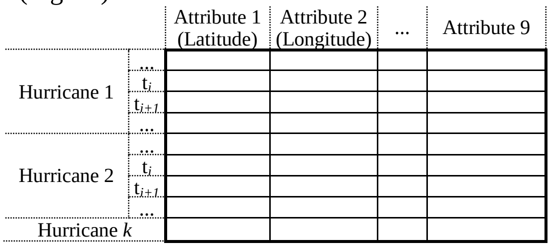
B. *Matrix of Locations' Sensor Data*

Secondly, properties of the *object* concept "*location*" are considered by looking into weather data at locations. In the dataset, there are about 20,000 sensor stations throughout The USA and, on average, each station has 5 sensors to measure weather phenomena. However, using all of the information will result in a large matrix with too many columns, so a reduction of the amount of columns is needed to reduce computing cost. Hence, we choose 9 attributes including: air-temperature, dew-point, precipitation, pressure, wind direction, wind speed, wind gust, relative humidity and distance to the hurricane's center, to which a storm is most relevant in our study context.

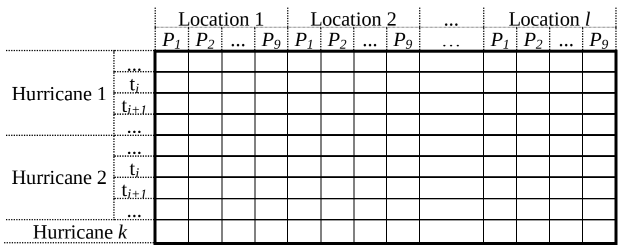
Moreover, as a location can have many weather stations, we assign the average value for each property. The distance between a county and hurricane's center is calculated as the average of distances of sensor stations in that county to the center. Each station's distance is measured as the shortest dis-tance of two points on the earth's surface by Haversine formula [13]. So in the designed model, a location has 9 columns of 9 properties, and a row vector of sensor data has 336x9 = 3,024 variables in total. With m_k data rows of a hurricane (exactly corresponding to rows in A), its data matrix is of m_kx3,024 dimension. Aggregating matrices of different hurricanes by row expansion results in the *sensor data matrix* (matrix **B**) as shown in Fig. 3b.

C. *Storm Damage Index Matrix*

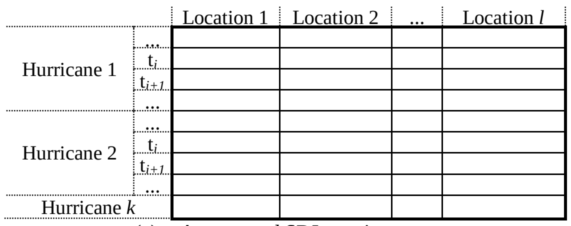
Thirdly, data matrix of the *predicate* "*causes-damage-to*" is formed from the SDI values, calculated by formula (5). Values of the storm-level and the storm-damage-level are collected from the Tropical Cyclone database [14], in which each of the two parameters is divided into 5 ranges. While the storm-level is relevant to wind-speed, the storm-damage-level is computed with respect to financial damage, human deaths and injuries. Value of the population rate is picked from the population data of The USA [15]. From the referred data, we set the damage level ranged from 0.0–5.9 and the population rate ranged from 0.0–0.9. Since there are 336 locations, the matrix has 336 columns to indicate damage index of those locations at a time T_i. Again, this time instance must exactly correspond to that of matrix **A** and matrix **B**. An aggregation of SDI matrices of different hurricanes results in the SDI matrix **C** (Fig. 3c).



(a) *Aggregated observation data matrix.*



(b) *Aggregated sensor data matrix.*



(c) *Aggregated SDI matrix.*

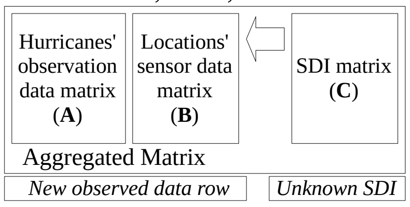
Fig. 3. *Matrices of subject, object and predicate components.*

D. *Aggregated Matrix For Multivariate Regression*

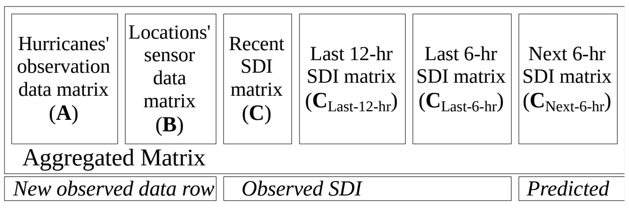
One may easily see that there is a correlation between co-variate data in matrices A and B with dependent variables in matrix C (on the same row index). This means that if new data rows (new row index) of A and B can be given, the correspon-ding data in C can be implied by a regression process. In fact, SUNS approach allows an aggregation of data matrices horizontally (column-expansion) to enable the multivariate regression on that matrix to find missing information.

Fig. 4a shows that aggregated matrix, which is ready for learning, with 16,575 rows and 3,369 columns. In term of revealed entries, it has 3,798,542 items (from A and B) and 16,890 SDI entries (from C), giving a density at ~6.83% which indicates that this is a very sparse matrix. However, we observe that there are about 1,656 wrong data items such as negative rainfall or abnormal too big values (that never appear in the reality). These items are removed by simply setting to value 0 (the sparsity is not affected). Finally, we normalize all columns to the range 0.0–1.0 based-on each column's meaning.

However, we are not going to do regression on the mention SDI matrix C, as it presents damages at the same time with (or more correctly, shortly after) the time when observation and sensor data are recorded. Our target is to make prediction for future SDI, firstly for next 6 hours. Therefore, we form another matrix $C_{Next\text{-}6\text{-}hr}$ as the same way of constructing C, but taking SDI value of next-6-hr for the time instance T_i. In addition, we recognize that historical damages of a hurricane can contribute to the prediction of its future states, as well as distinguish its pattern with others' patterns. Hence, we form two SDI matrices $C_{Last\text{-}6\text{-}hr}$ and $C_{Last\text{-}12\text{-}hr}$ for recording the past of damages of locations at 6 hours and 12 hours, respectively. Aggregation of these matrices with the previous one is given in Fig. 4b. In the end, the matrix has dimension of 16,575x4,377.


Hurricanes' observation data matrix (A)
Locations' sensor data matrix (B)
SDI matrix (C)
Aggregated Matrix
New observed data row
Unknown SDI

(a)


Hurricanes' observation data matrix (A)
Locations' sensor data matrix (B)
Recent SDI matrix (C)
Last 12-hr SDI matrix ($C_{Last\text{-}12\text{-}hr}$)
Last 6-hr SDI matrix ($C_{Last\text{-}6\text{-}hr}$)
Next 6-hr SDI matrix ($C_{Next\text{-}6\text{-}hr}$)
Aggregated Matrix
New observed data row
Observed SDI
Predicted

(b)

Fig. 4. *Aggregated data matrices. In (b), the row on the bottom indicates that "New observed data row" and "Observed SDI" are co-variate variables (corresponding to 5 matrices on the left), while "predicted" dependent variables are corresponding to the last matrix on the right.*

V. EXPERIMENT

As the comprehensive matrix for regression in our model has more than 16,000 rows and 4,000 columns, computing cost of iterating multiplication is high. For a reasonable real-time computation (to satisfy the prediction requirements), there is a strongly need of High Performance Computing resources, and this will be achieved in the future stage of the study. At the moment, our idea is tested with small configuration (smaller dimensions of data matrix and on personal computer), in which we believe that can provide sufficient evaluation of the model's structure, but not its calculating performance.

The testing program is implemented in C programming language, using SVDLIBC library (version 1.4) [16] for facto-rizing matrix into 3 lower rank components as in formula (1). The program allows adding one or many new rows at a time to the train-data matrix.

A. *Testing Scenarios*

So far, our preliminary testing strategy is based-on a natural expectation of the real life: *when a storm is going to come, predict its potential damages*. Therefore, data of the 4 first hurricanes (Charley, Katrina, Wilma and Gustav) are learned to predict the last one (Ike). As mentioned above, we relax both train-set and test-set to smaller configuration, and choose 12 hours (24 rows) of the peak time of hurricane Ike for testing. Be clear that in this context, one knows a storm has appeared somewhere else through other weather forecasting resources.

To decrease the dimensions of the train-set and test-set, we make a pruning procedure on rows and columns. *Firstly*, we choose time interval is 30 minutes, meaning that just data rows with minute value of timestamp is 0 or 30 are taken (e.g.: time instances 1:00 AM or 1:30 AM will be selected, but not 1:15 AM). *Secondly*, in matrix A we keep only columns that have more than a half of entries are revealed. *Thirdly*, we remove locations (in matrix C and the derived historical and future matrices) which are affected less than 10 times, and remove those locations' corresponding columns in matrix A too. As a result, there remain 54 locations and totally 328 columns of co-variate data. The training matrix has dimensions of 648x382 with 106,545 revealed entries (density ~43%), and the testing matrix has dimensions of 24x382 and 3677 revealed entries (density ~40.1%). Moreover, there are totally 1,296 regression values with 348 non-zero entries in the testing matrix.

Regarding that training matrix, we choose the reduced-rank equal to 190, which we consider as a significant reduction of dimensions, after observing that it gives the lowest root mean square error and relative error and is lower than a half of the number of columns. In addition, we set the λ parameter at 0.1 for RRPR and number of iterations of both algorithms at 100. Based-on this configuration, we exercise three test-cases to compare the predicting scenarios, and the performance of Naive-SVD and RRPR.

B. *Results*

Three test-cases are executed as below:

- In the first test-case, we simply associate all rows of the testing matrix into the training matrix to form the regression matrix. 348 non-zero entries (of the 1,296 regression values) are all set to 0 before the regression process, to indicate that they are missing values that need the algorithms to predict.
- In the second test-case, we alternatively combine each row of the testing matrix with the training matrix to repeatedly perform the regression process, without updating the training matrix. Note that non-zero dependent variables which should be predicted are all set to be 0 too.
- The third test-case is exercised similarly to the second one, but after each testing row is completed (and its real pattern is known), it will be added to the training matrix for updating the learning data. This tactic of constructing training data allows the model to learn new pattern(s) every time new streaming data of the current hurricane arrive. This execution is expected to perform better than the other cases, because it updates its "insights" through time.

1) *Training error*

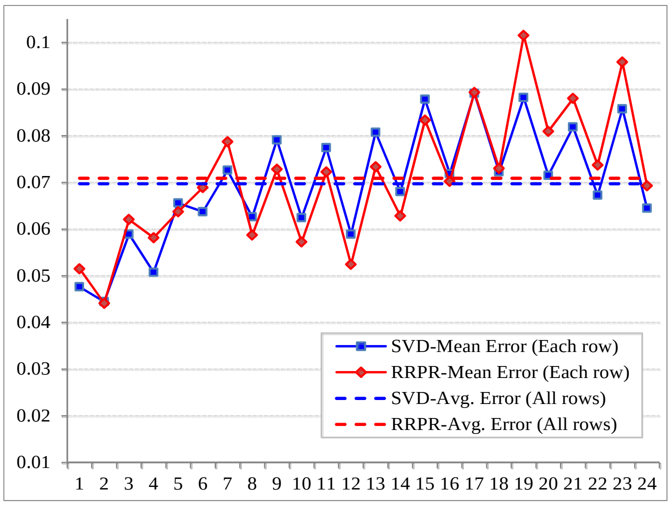
Table I lists the errors of the training process in three test-cases in term of RMSE and RELERROR over the set of known entries. As the resulted values indicate, "Naive" SVD produces the approximated matrix closer to the learned data than RRPR can do, with both RMSE and RELERR values are more than 2 times lower than those of RRPR in three tests. This means that "Naive" SVD "learns" better than RRPR does with the current data set and configuration. In term of test scenarios, "Naive" SVD performs best with the first row-expansion tactic, then the second case and finally the third case. This can be explained as the first case has least number of known entries. Contradictori-ly, and interestingly, RRPR does this best in the third case.

TABLE I. ROOT MEAN SQUARE ERROR AND RELATIVE ERROR

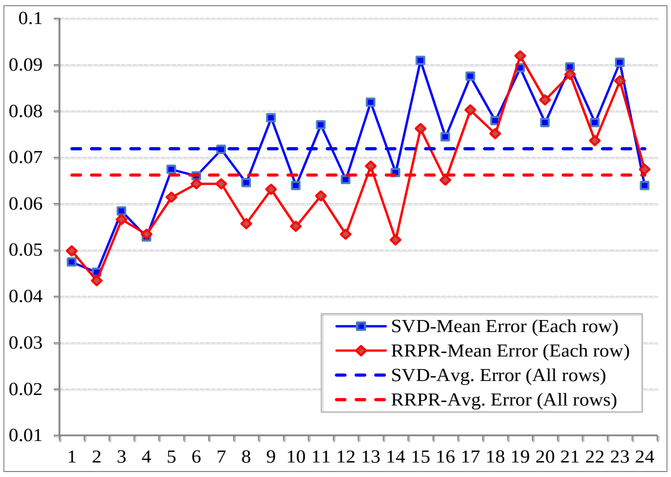
Test-case	"Naive" SVD		RRPR	
	RMSE	*RELERR*	*RMSE*	*RELERR*
Test-case 1	0.001427	0.004180	0.003117	0.009129
Test-case 2	0.001444	0.004213	0.003144	0.009174
Test-case 3	0.001449	0.004236	0.003106	0.009083

2) *Average error on test set*

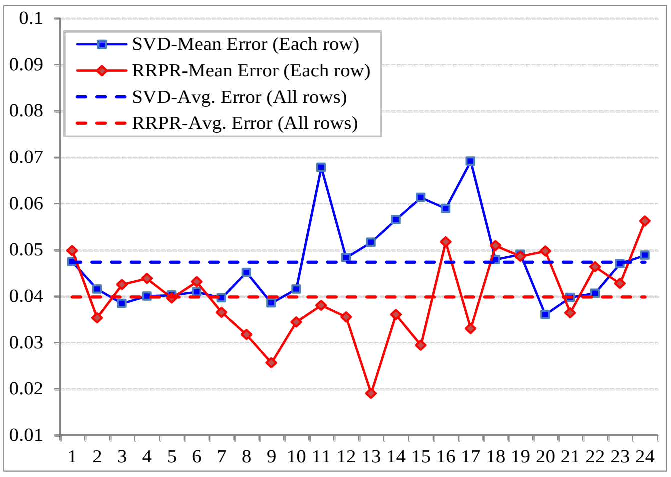
Since the values of RMSE and RELERR seem to be very diverse and not reflect clearly the patterns of errors, we use the average value of error of each tested row and the whole set of tested entries to analyze predicting quality. We denote "mean error" for each of 24 rows and "average error" for the whole set. Fig. 5 summarizes those values for the three tests, with "Naive" SVD is colored in red and RRPR is colored in blue.



(a) 1st test-case: Mean error of each row and Average error of all rows



(b) 2nd test-case: Mean error of each row and Average error of all rows



(c) 3rd test-case: Mean error of each row and Average error of all rows

Opposite to the errors over the known co-variate entries, errors over predicted dependent variables show different trends in three test-cases. While in the first one, “Naive” SVD seems to still predict better than RRPR (average error is ~0.697 com-paring to ~0.710, respectively), the second and third cases see better predictions of RRPR. RRPR's average error is ~0.066 in the second test (in comparison to ~0.072 of “Naive” SVD) and significantly reduces to ~0.040 in the third test (in comparison to ~0.047 of “Naive” SVD). This means that when it is able to learn and predict gradually, RRPR does better, especially when an online learning strategy is executed. Moreover, as variables are ranged from 0.0–1.0, the average error converges at 0.04 of RRPR can be considered significantly small and acceptable.

In addition, in the two former test-cases, patterns of mean error of both algorithms are quite similar, and look similarly in the two cases. Errors in these two scenarios have the trend to increase considerably when the new hurricane continues to generate more situations of happening and damages. Whilst in the last case, when training data is updated, mean error patterns fluctuate among rows, with RRPR keeps better stability than “Naive” SVD. This indicates that RRPR can better recognize new and “strange” patterns, even though “Naive” SVD also achieves this ability in the updating context.

And finally, we realize that while “Naive” SVD can approximate the model matrix to the revealed set better than RRPR, it is not as good as RRPR in making prediction. This can point out that RRPR is better in dealing with the over-fitting issue. These abilities of RRPR are very meaningful as we know no new storm is the same with other previous ones.

VI. CONCLUSION

The paper has presented our idea in adapting and extending the SUNS framework for applying to the hurricane disaster damages forecasting problem, taking advantages of existing SW resources. We have described an innovative technique for converting streaming RDF data of different hurricanes into one data matrix which provides the input of multivariate regression for learning and predicting. Especially, data are collected from different sources and may contain noises or be not consistent, but still can be combined together by column-expansion and row-expansion procedures. In the model, not only sensor and observation data at a time-point but also historical patterns of SDI damage at previous 12 hours and 6 hours are used to imply the output, the SDI values at next 6 hours. Moreover, two MC algorithms have been applied successfully to perform the reduced-rank regression process.

At this stage, we have achieved tests on small amount of data, with some interesting results come out from three testing scenarios. Despite the fact that learned data are incomplete and inconsistent, the model can still reflect the pattern of damages of 4 hurricanes and estimate future SDI values of a new one with reasonable small error. The error trends of two algorithms over train-data and test-data are various throughout the three test-cases. In general, RRPR does better than “Naive” SVD in term of prediction, even though “Naive” SVD can approximate the model matrix closer to the data matrix. In addition, the third test-case figures out that when data is updated in a streaming manner, the quality of prediction can be improved significantly for both algorithms.

VII. DISSCUSSION AND FUTURE WORK

*Suggestion***.** Our achievement suggests a promising model for bridging the seemingly undiscovered gap between two research areas, SW technologies and storm disaster damages prediction, and recommends an alternative to, or support to, meteorological systems to deal with the challenge of weather extreme event prediction.

Extendable abilities. Our model can be combined with other numerical forecasting methods and/or data to predict many kinds of damages (in terms of severity or human, asset and economic losses), for a wide range of scale of locations (district, city, country, or region).

Weaknesses. Although the study has achieved some promising results of application of SUNS approach for estimating SDI value, there are some obstacles for our model. *Firstly*, as a statistical model, it just can deal with typical "learned" patterns and lacks the ability to deal with completely new situations. Therefore, it needs to be associated with other techniques (such as heuristics or meteorological models' results) to improve the forecasting performance. *Secondly*, the SDI value in our model can be improved, because when a storm is active over some regions, the damages it causes to locations close to its eye are often higher than other surrounding places. And *thirdly*, there seems to be no other similar works or benchmarks for having comparison and evaluation of our results.

Future work. For the next phase of the study, we are going to move forward on using the power of High Performance Computing for testing on the full configuration of data and making prediction for longer future, as well as improving the quality of the learning and predicting model. Furthermore, we aim at applying the suggested model for investigating disaster data of Republic of Korea (mainly focus on flood and land-slide), and combining with text data (such as Twitter) for a wider range of disaster damages prediction.

ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIP/IITP. [B0101-15-0559, Developing On-line Open Platform to Provide Local-business Strategy Analysis and User-targeting Visual Advertisement Materials for Micro-enterprise Managers].

REFERENCES

[1] R. D. Knabb, J. R. Rhome & R. D. Brown. Tropical cyclone report: Hurricane Katrina. Technical report, NOAA National Hurricane Center, 23–30 August, 2005.

[2] J. Burston, W. Daniel, and R. Tomlinson. The real-time needs of emergency managers for tropical cyclone storm tide forecasting: results of a participatory stakeholder engagement process. Natural Hazards, vol. 78, pp. 1653-1668, 2015.

[3] V. R. Durai, , S. D. Kotal, R. Bhowmik, and R. Bharadwaj. Performance of Global Forecast System for the prediction of intensity and track of very severe cyclonic storm 'Phailin' over North Indian ocean. In High-Impact Weather Events over the SAARC Region, pp. 191-203. Springer International Publishing, 2015.

[4] D. Zastrau, M. Schlaak, T. Bruns, R. Elsner, and O. Herzog. The relation between storm risk and wind and wave forecast accuracy in the North Atlantic ocean. International Journal of Environmental Science and Development, vol. 6, no. 2, 2015.

[5] Q. K. Tran, J. H. Um, S. K. Song. Matrix completion for storm damages prediction. International Semantic Web Conference (ISWC) 2015 Poster & Demo session (in press), 2015.

[6] Y. Zhang, Pham Minh Duc, O. Corcho & J. P. Calbimonte. SRBench: A streaming RDF/SPARQL benchmark. In: The Semantic Web–ISWC 2012, pp. 641-657, Springer Berlin Heidelberg, 2012.

[7] V. Tresp, Y. Huang, M. Bundschus & A. Rettinger. Materializing and querying learned knowledge. Proc. of IRMLeS, 2009.

[8] Y. Huang, V. Tresp, M. Bundschus, A. Rettinger & H. P. Kriegel. Multivariate prediction for learning on the semantic web. In: Inductive Logic Programming, pp. 92-104, Springer Berlin Heidelberg, 2011.

[9] X. Jiang, Y. Huang, M. Nickel & V. Tresp. Combining information extraction, deductive reasoning and machine learning for relation prediction. In The Semantic Web: Research and Applications, pp. 164-178, Springer Berlin Heidelberg, 2012.

[10] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein. Imputing missing data for gene expression arrays, 1999.

[11] R. Mazumder, T. Hastie & R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. Journal of Machine Learning Research, vol. 11, pp. 2287-2322, 2010.

[12] Linked Sensor Data and Linked Observation Data, http://wiki.knoesis.org/index.php/LinkedSensorData, Last accessed: September 2015.

[13] Hervasine formula, http://www.movable-type.co.uk/scripts/latlong.html, Last accessed: September 2015.

[14] Tropical Cyclone, https://www.ncdc.noaa.gov/billions/events, Last accessed: September 2015.

[15] Population of USA, https://www.census.gov/popest/data/intercensal/cities/files/SUB-EST00INT.csv, Last accessed: September 2015.

[16] SVDLIBC – TedLab – MIT, http://tedlab.mit.edu/~dr/SVDLIBC/, Last accessed: September 2015.