Studying the Impact of Partition on Data Reduction for Very Large Spatio-temporal Datasets

Nhien An Le Khac
School of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland
e-mail: an.lekhac@ucd.ie

Martin Bue
Ecole Polytechnique Universitaire de Lille
Villeneuve d'Ascq cedex, France
e-mail: Martin.Bue@polytech-lille.net

M-Tahar Kechadi
School of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland
e-mail: tahar.kechadi@ucd.ie

***Abstract*—Nowadays, huge amounts of data are being collected with spatial and temporal components from sources such as metrological, satellite imagery etc. Efficient visualisation as well as discovery of useful knowledge from these datasets is therefore very challenging and becoming a massive economic need. Data Mining has emerged as the technology to discover hidden knowledge in very large amounts of data. Furthermore, data mining techniques could be applied to decrease the large size of raw data by retrieving its useful knowledge as representatives. As a consequence, instead of dealing with a large size of raw data, we can use these representatives to visualise or to analyse without losing important information. Recently, we proposed a new approach based on different clustering techniques for data reduction to help analyse large spatio-temporal data. This approach is based on the partition of huge datasets due to the memory constraint. In this paper, we evaluate the impact of various numbers of partitions on our data reduction approach.**

Keywords-spatio-temporal datasets; data reduction; data partition; density-based clustering; shared nearest neighbours

I. INTRODUCTION

Many natural phenomena present intrinsic spatial and temporal characteristics. Besides traditional applications, recent concerns about climate change, the threat of pandemic diseases, and the monitoring of terrorist movements are some of the newest reasons why the analysis of spatio-temporal data has attracted increasing interest. With the recent advances in hardware, high-resolution spatio-temporal datasets are collected and stored to study important changes over time, and patterns of specific events. However, these datasets are often very large and grow at a rapid rate. So, it becomes important to be able to analyse, discover new patterns and trends, and display the results in an efficient and effective way.

Spatio-temporal datasets are often very large and difficult to analyse [1][2][3]. Fundamentally, visualisation techniques are widely recognised to be powerful in analysing these datasets [4], since they take advantage of human abilities to perceive visual patterns and to interpret them [5]. However, spatial visualisation techniques currently provided in the existing geographical applications are not adequate for decision-support systems when used alone. For instance, the problems of how to visualise the spatio-temporal multi-dimensional datasets and how to define effective visual interfaces for viewing and manipulating the geometrical components of the spatial data [6] are the challenges. Hence, alternative solutions have to be defined. Indeed, new solutions should not only include a static graphical view of the results produced during the data mining (DM) process, but also the possibility to obtain dynamically and interactively different spatial and temporal views as well as interact in different ways with them. DM techniques have been proven to be of significant value for analysing spatio-temporal datasets [7][8]. It is a user-centric, interactive process, where DM experts and domain experts work closely together to gain insight on a given problem. In particular, spatio-temporal data mining is an emerging research area, encompassing a set of exploratory, computational and interactive approaches for analysing very large spatial and spatio-temporal datasets. However, several open issues have been identified ranging from the definition of techniques capable of dealing with the huge amounts of spatio-temporal datasets to the development of effective methods for interpreting and presenting the final results.

Analysing a database of even a few gigabytes is an arduous task for machine learning techniques and requires advanced parallel hardware and algorithms. Huge datasets create combinatorially explosive search spaces for DM algorithms which may make the process of extracting useful knowledge infeasible owing to space and time constraints. An approach for dealing with the intractable problem of learning from huge databases is to select a small subset of data for mining [2]. It would be convenient if large databases could be replaced by a small subset of representative patterns so that the accuracy of estimates (e.g., of probability density, dependencies, class boundaries) obtained from such a reduced set should be comparable to that obtained using the entire dataset. This approach can be viewed as similar to sampling, a technique that is commonly used for selecting a subset of data objects to be analysed. There are different techniques for this approach such as the scaling by factor [9][10], data compression [11], clustering [12], etc. Recently, we proposed a reduction technique based on clustering [13] for large size of spatio-temporal datasets. Clustering is used on spatio-temporal data to take advantage of the fact that, objects that are close together in space and/or in time can usually be grouped together. As a consequence, instead of dealing with a large size of raw data, we can use these cluster

 ISBN:978-1-61208-115-1

representatives to visualise or to analyse without losing important information [13]. In this solution, multi-partition approach has been applied to cope with huge size of input datasets i.e., all draw dataset is divided in equal parts and each time, only one partition can be processed. Basically, the number of partitions depends on the runtime memory. We normally optimise the number of partitions by maximise the size of each partition in according to the size of memory. Indeed, in order to exploit efficiently the capacity of multithreading on the multi-core platforms, a larger number of partitions in smaller sizes is required. However, the changing of partition size could effect on final representatives because of changing the position of nearest neighbours. A best trade-off between the number of partitions and final representatives should be determined. To do so, in this paper, we study the impact of the number of partitions on final representative results.

The rest of this paper is organised as follows. In Section II, we present background related to this subject. Section III describes briefly our data reduction technique based on clustering techniques. We discuss on the impact of the number of partitions in Section IV and we evaluate then this impact on real large spatio-temporal dataset in Section V. Section VI is to deal with the conclusion and our future work.

II. BACKGROUND

In this section, we present firstly the spatio-temporal data mining system and then different clustering techniques applied for reducing spatio-temporal datasets.

A. Spatio-temporal data mining

Spatio-temporal DM represents the junction of several research areas including machine learning, information theory, statistics, databases, and geographic visualisation. It includes a set of exploratory, computational and interactive approaches for analysing very large spatial and spatio-temporal datasets. Recently various projects have been initiated in this area ranging from formal models [4][14] to the study of the spatio-temporal data mining applications [5][14]. In spatio-temporal data mining the two dimensions "spatial" and "temporal" have added substantial complexity to the traditional DM process. It is worth noting, while the modelling of spatio-temporal data at different levels of details presents many advantages for both the application and the system. However it is still a challenging problem. Some research has been conducted to integrate the automatic zooming of spatial information and the development of multi-representation spatio-temporal systems [15][16][17]. However, the huge size of datasets is an issue with these approaches.

In [8][9], the authors proposed a strategy that is to be incorporated in a system of exploratory spatio-temporal data mining, to improve its performance on very large spatio-temporal datasets. This system provides a DM engine that can integrate different DM algorithms and two complementary 3-D visualisation tools. This approach reduces their datasets by scaling them by a factor F; it simply runs through the whole dataset taking one average value for the F^3 points inside each cube of edge F. This reducing technique has been found to be inefficient as a data reduction method which may lose a lot of important information contained in the raw data.

B. Data reduction by clustering

To the best of our knowledge, there is only our recently work [12][13] which proposed a clustering-based data reduction method in the context of analysing spatio-temporal datasets. This method is based on clustering [1] to cope with the huge size of spatio-temporal datasets in order to facilitate the mining of these datasets. The main idea is to reduce the size of that data by producing a smaller representation of the dataset, as opposed to compressing the data and then uncompressing it later for reuse. The reason is that we want to reduce and transform the data so that it can be managed and mined interactively. Clustering technique used in this approach is K-Medoids [12] and density-based [18]. The advantage of these techniques is simple in term of basic algorithms used. In the first technique, its representatives (medoids points) cannot however reflect adequately all important features of the datasets because it is not sensitive to the shape of the datasets (convex) as the second technique does with its specific core points [20] used as representatives. A brief presentation of this second technique is in the following section.

III. DENSITY-BASED DATA REDUCTION

As described in [8][9], our spatio-temporal data mining framework consists of three steps: data preparation, data mining and visualisation. The visualisation step contains different visualisation tools that provide complementary functionality to visualise and interpret mined results. Normally, the raw spatial-temporal dataset is too large for any algorithm to process; the goal of the data preparation step is to reduce the size of that data by producing a smaller representation of the dataset, so-called representatives, without losing any relevant information, as opposed to compressing the data and then uncompressing it later for reuse. Furthermore, we aim to reduce and transform the data so that it can be managed and mined interactively. It allows the mining step to apply mining technique such as clustering, association rules on the representatives (tightly grouped data objects) to produce new knowledge and ready for evaluation and interpretation.

We proposed a new data reduction method based on clustering to help with the mining of the very large spatio-temporal dataset [13]. The use of cluster representatives helps to filter (reduce) datasets without losing important/ interesting information because clustering techniques group data objects based on the characteristics of the objects and their relationships. It aims at maximising the similarity within a group of objects and the dissimilarity between the groups in order to identify interesting structures in the underlying data. Concretely, we have implemented a combination of a density-based clustering - a modification of DBSCAN algorithm [18]; and a graph-based clustering - a Shared Nearest Neighbor Similarity (SNN) algorithm [19], in this approach. The advantage of this combination has been

 ISBN:978-1-61208-115-1

discussed in [13]. Briefly, this combination is not only efficient with spatial datasets as it takes into account the shape (convex) of the data points but also addresses the problems of low similarity and differences in density. The whole can be resumed as follow: (1) a pre-processing is applied on raw datasets to filter NULL values. (2) SNN similarity graph is built for all datasets. Similarity degree of each data object is also computed in this step. The two parameters *Eps* and *Minpts* are selected based on these similarity degrees. Next, a DBSCAN-based algorithm is carried out on the datasets to determine *core objects, specific core objects, density-reachable objects and density-connected objects*. The definition of these objects are defined in [18][20]. *Core objects* or *specific core objects* are selected as cluster representatives that form a new (meta-) dataset (3). This dataset can then be analysed and produce useful information (i.e., models, rules, etc.) by applying other DM techniques (the mining step). It is important to note that data objects that have a very high similarity between each other can be grouped together in the same clusters. As a result of this step, the new dataset is much smaller than the original data without losing any important information from the data that could have an adverse effect on the result obtained from mining the data at a later stage. Fig. 1 and Fig. 2 show respectively all data points before and after this density-based clustering approach on around 25 million data points of 4 dimensions X, Y, Z, QCLOUD for the time step 2 of the Isabel hurricane datasets [21]. The representatives (Fig. 2) could reflect the general shape of hurricane based on (X,Y,Z,QCLOUD) comparing to their whole original points (Fig.2). More results and discussion for other time-steps can be found in [13].

Memory issue. The sizes of spatio-temporal datasets are normally very large. For instance, Hurricane Isabel dataset [21] is represented by a space of 500 × 500 × 100 with 25 × 10^6 data points (data objects). Therefore, a dissimilarity matrix (used to determine the distance between any two data points in the dataset) of all data points would take $25^2 \times 10^{12}$ × 6 bytes. It exceeds the memory capacity of commodity computation machines. Normally, tree-based topologies such as R-tree [22] have been applied to index data points in order to tackle this memory issue. However, high frequency of queries on a huge tree structure and of secondary memory access would be a performance impact. By using SNN algorithm in our approach, we only need to keep the distances to k-nearest neighbour of each data point. For instance, for Hurricane Isabel dataset presented above, we would need 25 × 10^6× 6 × k bytes of memory for storing k-nearest information of all data points.

IV. STUDYING THE IMPACT OF THE NUMBER OF PARTITIONS ON REDUCTION RESULTS

In our approach, if the computational variables of the whole space cannot be all loaded into main memory, a local-global solution will be applied. In this solution, a whole dataset is equally divided in un-joint partitions. The number of partition depends on the memory size where one partition can be processed at each time. Let D, P_i, M, S_P be all raw datasets, partition i, memory reserved for this computation and computational memory needed for each partition respectively, this problem can be officially defined as:

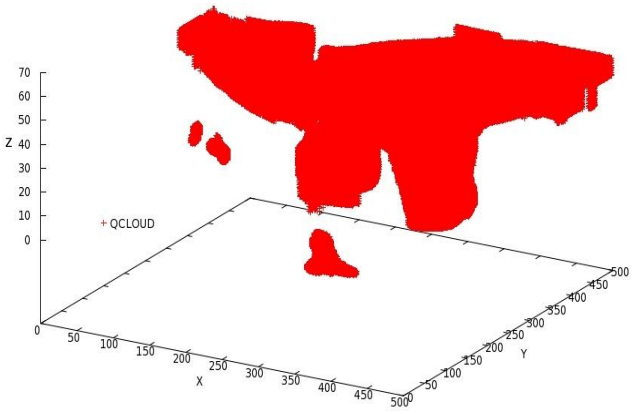

Figure 1. All datasets for QCLOUD, timestep 2.

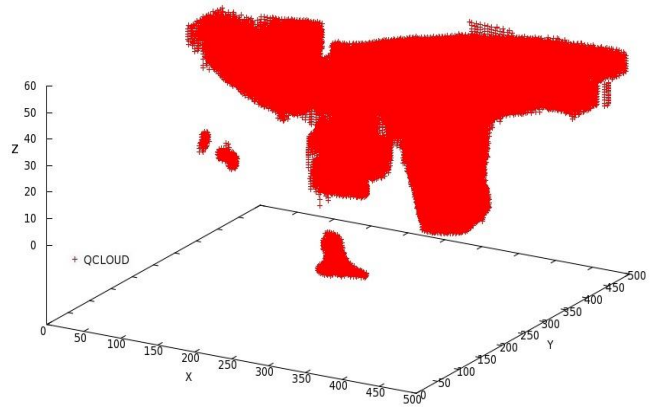

Figure 2. Data reduction by SNN-DBSCAN for QCLOUD, timestep 2.

$$D \equiv \bigcup_{i=1}^{n} P_i, \forall i, j \in \{1..n\}, i \neq j : P_i \cap P_j = \varnothing \quad (1)$$

$$M \equiv S_p \quad (2)$$

For each partition P_i, we apply our algorithm described above to determine *core objects* C_{Pi}, *specific core objects* SC_{Pi} and *clusters* Cl_{Pi}. As mentioned above, both *core objects* and *specific core objects* can be used as representatives. However, according to the experiments [13], the ratio between the number of core objects and the number of all raw data objects is relative high (~90%). Thus, only *specific core objects* are considered *local representatives* and clusters Cl_{Pi} are called *local clusters*. After processing all partitions, there are two methods to build the *global clusters*:

- merging local clusters together based on their core objects and the distance ε between core objects located on the borders of difference partitions to build global clusters and the global representatives (union of all local representatives). This method is

simple. However, the determination of ε is an issue because of the variety in density of different partitions. Indeed, our reduction approach is based on SNN degree [19], not on the distance.

- all local representatives will be joined together to create a new dataset. A DBSCAN-based algorithm will be carried out on this new dataset to create new clusters called *global clusters*. The *core objects* are *global representatives*. Our studying is focusing on this method.

The *global representatives* of *global clusters* form a reduction dataset. In both methods above, the number of partition *n* is an important impact on the final reduction results. When we increase *n*, the number of local representatives is also increased and that affects to the global results, especially the ones locate near the border of each partition because most of them would not be representatives at the global level. Basically, the smaller number *n* is expected in according to the constraint of the running memory and in the optimal case $M \equiv S_{Pi}$ (cf. (2)) i.e., only one partition is loaded with its maximum size in according to the computational memory reserved.

Meanwhile, today, running efficiently applications on multi-core platforms is a challenge as the multi-core hardware is widely used; the software has not been ready yet. In the context of analysing huge size of spatio-temporal datasets, the multi-core programming is a solution for improving the processing time. To do so, these current approaches should be designed in multicore-ready style i.e., it allows to exploit the maximum capacity of multithreading on the target system. Concretely, in our approach, multithreading can be applied in two different methods. In the first one, multithreading is carried out on one partition P_i loaded in the memory. In this case, important changes are needed for the algorithm in according to multithreading environment. In the second method, multi partitions can be loaded in the memory. This case can efficiently benefit the capacity of multithreading without changing the algorithm. In this case:

$$M \equiv \bigcup S_{P_j}, P_j \in D, Count(S_{P_j}) = m \qquad (3)$$

However, increasing the number of partitions would lead to a performance issue in term of final reduction results as discussed above. So, we should determine the optimal *m* i.e., the value maximum of *m* that does not affect the final results. On the other hand, a large number of partitions is also a running time issue as most of processing time is reserved for loading/unloading datasets from/to computational memory instead of calculating.

V. EVALUATION

In this section, we study study the impact *m* (c.f. (3)) with real spatio-temporal datasets in the context of data reducing by using DM techniques described in Section III. The dataset is the Isabel hurricane data [21] produced by the US National Centre for Atmospheric Research (NCAR). It covers a period of 48 hours (time-steps). Each time-step contains several atmospheric variables. The grid resolution is 500×500×100. The total size of all files is more than 60GB (~ 1.25 GB for each time-step). The experimentation details and a discussion are given below.

The platform of our experimentation is a PC of 3.4 GHz Dual Core CPU, 1GB RAM using Java 1.6 on Linux kernel 2.6. Datasets of each time-step include 13 non-spatio attributes, so-called dimensions. In this evaluation, QCLOUD is chosen for analysis; it is the weight of the cloud water measured at each point of the grid. The range of QCLOUD value is [0…0.00332]. The chosen time-step is 2 as the different time steps is similar in term of processing. We also filter the NULL value and land value of testing dataset. Totally, the testing dataset contains around 25 million data points of 4 dimensions X, Y, Z, QCLOUD for each time step. After the reduction process, the number of data objects is around 100000. Due to the memory constraint, the number of partitions is varied from 40 to 180.

Figures from 3 to 7 show the clustering results for 40, 70, 100, 120 and 180 partitions respectively. In each figure, we only show 9 biggest clusters with different colours[1]. Other clusters are in the same colour: black-(9). Table 1 gives more details on the number of clusters, the number of representatives (specific core objects) for each case of partition. By observing these figures, we recognise that:

- the three biggest clusters (blue-(0), red-(1) and green-(2)) are similar in all cases i.e., they are not strongly affected by the number of partitions. This also means that if the size of a cluster is large enough it then can preserve it shape against a reasonable number of fragmentations. The optimal relationship between the size of cluster and this number is out of scope of this paper.
- small difference in the number of clusters and their shapes among cases of 40, 70, 100 and 120 partitions . In the 40-partition case (Fig. 3), the 9th cluster (grey (8)) is clearer than the rest. The reason is that it gets a smaller fragment degree of raw data than other cases.
- The three cases 70, 100 and 120-partition are similar in terms of cluster shape and cluster position.
- Size and shape of clusters from the 4th one (yellow-(3)) of the 180-partition case is different compared to other cases. This means that this number of partitions affects on the final results.

Besides, as shown in the Table 1, the number of clusters created by all cases is similar with the difference is less than 5% (varied from 82 clusters to 87 clusters). Indeed, the number of representatives (specific core objects) is decreasing when we increase the number of partitions. Because there is an increasing of local representatives, that are not global representatives, in each partition. However this

[1] We use different colours rather than makers ('+','*', 'x'...) to distinguish different clusters because of the large number of plotting points (~ 100000). We put however the number (0, 1, 2...) on each cluster to make it easier to distinguish with other clusters in the case of black & white hardcopy.

 ISBN:978-1-61208-115-1

difference among these numbers is quite small (less than 1%).

Basing on these observations, we can vary the number of partitions loaded in the computational memory at the same time to exploit efficiently the multithreading capacity offered by multi-core platforms. Concretely, as shown in this experiment, we should start at 40-partition case with one partition loaded due to the memory constraint; we can obviously load from 2 to 4 partitions at the same time with a minor effect on the final results.

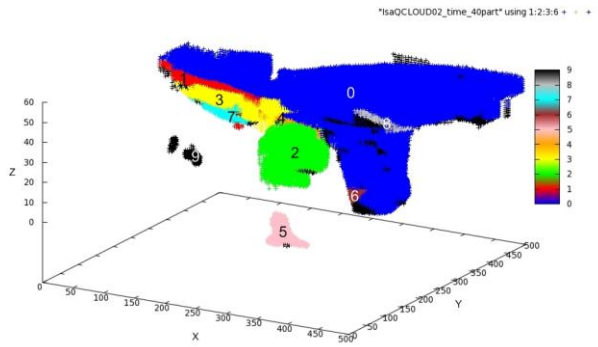

Figure 3. 40-partition.

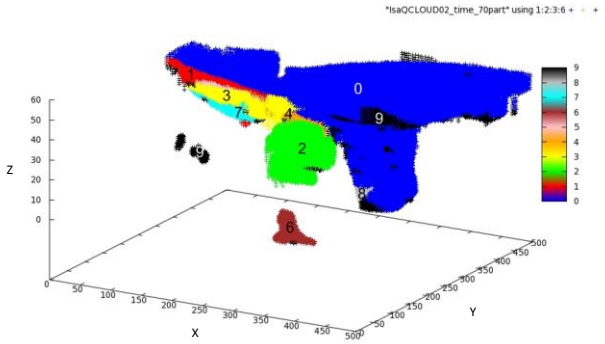

Figure 4. 70-partition.

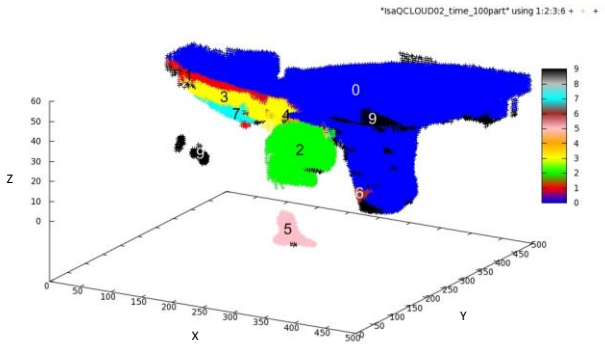

Figure 5. 100-partition.

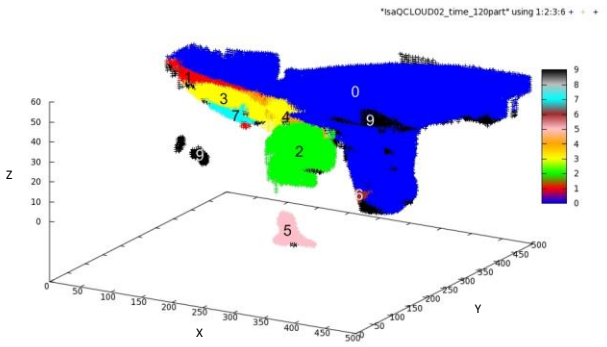

Figure 6. 120-partition.

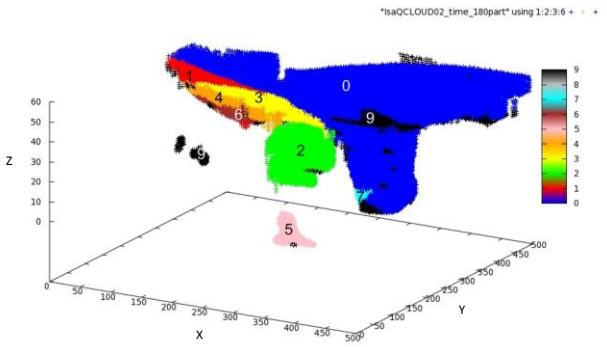

Figure 7. 180-partition.

As a brief conclusion, these experiments above show that we can process more partitions at the same time to benefit the multi-core environment in order to increase the speed-up of processing time.

TABLE I. REPRESENTATIVES AND CLUSTERS OF DIFFERENT NUMBER OF PARTITIONS

Number of partitions	Number of representatives	Number of clusters
40	103422	82
70	103139	84
100	102901	87
120	102708	86
180	102295	83

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we study the impact of the number of partitions on final representative results in the context of using clustering techniques for reducing the large size of spatio-temporal datasets. As there is a limitation of main memory, then the multi-partition approach has been applied.

 ISBN:978-1-61208-115-1

Basically, a minimum number of partitions is expected to benefit the memory capacity as well as to preserve the final results. Besides, in order to exploit efficiently multi-core platforms, more partitions need to be processed at the same time. Thus, an optimal number of partitions should be determined. The experimental results for QCLOUD of the Isabel hurricane in its space X,Y,Z for one time-step show that we can increase the number of partition with a smaller size loaded in the computational memory without effecting the final results. In the case where we should tackle with huge size of datasets, if the speedup is increasing from 3 to 4 times then it is an important gain in term of processing time.

A more extensive evaluation is on-going. In the future we intend to analyse different combinations of dimensions over more time steps to determine an optimal number of partitions for all cases. Indeed, we also carry out tests with multithreading techniques on multi-core platforms to evaluate the speedup gain as well as to prove the robustness of our approach of reducing spatio-temporal datasets.

REFERENCES

[1] Dunham, M. H., Data Mining: Introductory and Advanced Topics, Prentice Hall, 2003

[2] Tan, P-N., Steinbach, M. and Kumar, V., Introduction to Data Mining, Addison Wesley, 2006

[3] Ye, N. (ed), The Handbook of Data Mining. Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey, USA 2003

[4] Johnston W.L., "Model visualisation, in: Information Visualisation in Data Mining and Knowledge Discovery", Morgan Kaufmann, Los Altos, CA, 2001, pp. 223–227.

[5] Andrienko N., Andrienko G., and Gatalsky P., "Exploratory Spatio-Temporal Visualisation: an Analytical Review", Journal of Visual Languages and Computing, special issue on Visual Data Mining. December, v.14 (6), 2003, pp. 503-541.

[6] Liu, H. and Motoda, H., On Issues of Instance Selection, Data Mining Knowledge Discovery 6, 2, April, 2002, pp.115-130.

[7] Roddick, J. F., Hornsby, K., and Spiliopoulou, M., An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. Proceedings of the First International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining-Revised Papers, September 12, 2000, pp.147-164.

[8] Roddick, J.F. and Lees, B.G., Paradigms for Spatial and Spatio-Temporal Data Mining, In Geographic Data Mining and Knowledge Discovery. Miller H. and Han J. (Eds), Taylor & Francis, 2001

[9] 9. Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., and Kechadi, T., Exploratory Spatio-Temporal Data Mining and Visualization. Journal of Visual Languages and Computing, 18, 3, June, 2007, pp.255-279.

[10] Bertolotto, M., Di Martino, S., Ferrucci, F., and Kechadi, T., Towards a Framework for Mining and Analysing Spatio-Temporal Datasets, International Journal of Geographical Information Science, 21, 8, July, 2007, pp.895-906.

[11] Sayood, K., Introduction to Data Compression, 2nd Ed., Morgan Kaufmann, 2000

[12] Whelan, M., Le-Khac, N-A. and Kechadi, M-T., Data Reduction in Very Large Spatio-Temporal Data Sets, IEEE International Workshop On Cooperative Knowledge Discovery and Data Mining 2010 (WETICE 2010), Larissa, Greece, June 2010

[13] Le-Khac, N-A., Bue, M., Whelan, M., and Kechadi, M-T., A clustering-based data reduction for very large spatio-temporal datasets, The 6th International Conference on Advanced Data Mining and Applications (ADMA2010), Springer Verlag LNAI, November 19-21, 2010, ChongQing, China (to appear)

[14] Costabile M.F., Malerba D. (Editors), Special Issue on Visual Data Mining, Journal of Visual Languages and Computing, Vol. 14, December, 2003, pp.499-510.

[15] Bettini, C., Dyreson, C.E., Evans, W.S., Snodgrass, R.T., A glossary of time granularity concepts. In Proceedings of Temporal Databases: Research and Practice, Lecture Notes in Computer Science, Vol. 1399, Springer-Verlag, 1998, pp. 406–413.

[16] Bettini, C., Jajodia, S., Wang, X., Time Granularities in Databases, Data Mining, and Temporal Reasoning, Springer-Verlag, 2000

[17] Cattel, R., et al., The Object Database Standard: ODMG 3.0, Morgan-Kaufmann, 1999

[18] Ester, M., Kriegel, H.-P., Sander, J.,Xu, X., A Density-Based Algorithm for Discovering clusters in Large Spatial Databases with Noise. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), pp.226-231, Portland, OR, USA, 1996

[19] Jarvis, R. A. and Patrick, E.A., Clustering using a similarity Measure Based on shared Nearest Neighbours. IEEE Transactions on Computers, C-22(11), 1973, pp.1025-1034.

[20] Januzaj, E.,Kriegel, H-P., Pfeifle, M., DBDC: Density-Based Distributed Clustering. Proc. 9th Int. Conf. on Extending Database Technology (EDBT), Greece, 2004, pp.88-105.

[21] National Hurricane Center, Tropical Cyclone Report: Hurricane Isabel, http://www.tpc.ncep.noaa.gov/2003isabe l.shtml, 2003

[22] Guttman, A., R-Trees: A Dynamic Index Structure for Spatial Searching. Proc. ACM SIGMOD International Conference on Management of Data, 1984, pp. 47–57.

 ISBN:978-1-61208-115-1