BAJOPAS Volume 13 Number 1, June, 2020
http://dx.doi.org/10.4314/bajopas.v13i1.13
Bayero Journal of Pure and Applied Sciences, 13(1): 90 - 96
Received: April, 2020
Accepted: June, 2020
ISSN 2006 – 6996

FINDING BANDED PATTERNSIN LARGE DATA SET USING SEGMENTATION

Abdullahi[1], F. B and Coenen[2], F
[1]Department of Computer Science, Ahmadu Bello University Zaria Kaduna-State
[2]Department of Computer Science, University of Liverpool United Kingdom
Email: fbabdullahi@abu.edu.ng Phone: 08039235984

ABSTRACT
This paper presents a mechanism for finding banded patterns on largezero-one N-Dimensional data using segmentation technique. Traditionally, banding problem requires the generation of permutations. In this paper, Banded Pattern Mining (BPM) algorithms have been used, the BPM approximate and BPM exact. BPM algorithm incorporate abanding Score mechanism that does not consider large number of permutations. Although these algorithms operates well in sizable N-D datasets, large N-D dataset that cannot be stored easily on computer internal memory still present a challenge.To this end, a segmentation technique for discovering banding in large data compatible with the BPM is proposed. The technique was evaluated using a real life datasets the Great Britain (GB) cattle tracing system that represents the movements of all cattle in GB. From the reported evaluations, the mechanism was able to identify bandings in zero-one N-D datasets using series of data segmentstaken from a large N-D datasetwithin shortest possible time. The result shows BPM exact algorithm as the most effective in terms of overall banding result and BPM approximate algorithm as the most efficient in terms of runtime.
Keyword: Banded Pattern Mining, Large data, N-Dimension, Data Segmentation.

INTRODUCTION

This paper presents mechanism for finding banded patterns in large zero-one N-Dimensional (N-D) data (large in this context is interpreted as data not easily held in the computer internal memory). For better understanding, this paper presents one entries as "dots" and zero entries as "empty space". A banded Pattern in zero-one N-Dimensional data space is one where we rearrange the dimensions so that the nonzero entries are presented about the diagonal. The objective of this paper is to investigate and evaluate segmentation technique compatible with the BPM mechanism that do not require consideration of large numbers of permutations but use series of data segments to find banded patterns in N-D data. Figure 1; present an example of 3-Dbanding configuration with cells holding one or more dots.

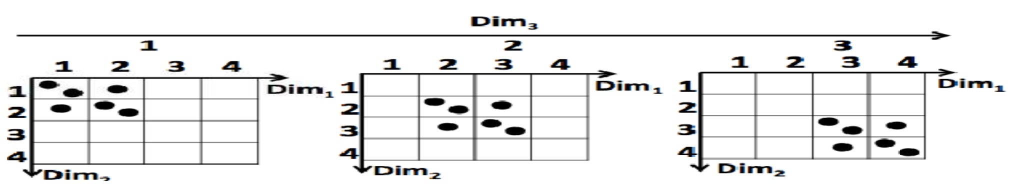


Figure1: 3-D configuration featuring a banding with one or more dot

The contribution of this paper are: (i) to identify approximate and exact bandings in large zero-one dataset that features one or more dots per location, (ii) the technique to apply banding on large dataset using segmentation and (iii) the application of banding to real life dataset (the GB cattle datasets). Traditionally, the banding problem is consider in terms of a number of permutations and the algorithm works only on 2D data (Gemma et al, 2008; Gemma et al, 2011; Juntilla, 2011). Mannila and Terzi in 2007, proposed nestedness and segmented nestedness of binary matrix; they defined the concept of k-nestedness, as a matrix is (almost) k-nestedness if the sets of columns can be partition into k-parts so that each parts is almost nested. They computed how far a matrix is from being k-nestedness by finding a good partition of columns into k-parts.

Reordering a binary matrix is NP-Complete problem (Cheng, 1973;Papadimitriou, 1976; Cuthill and McKee, 1969) To avoid the need for testing and generating of permutation, this paper present a segmentation technique compatible with the BPM algorithm and incorporate the possibility of some cells in the matrix to hold one or multiple dots. In Abdullahi et al (2014a), the author proposed a 2D-BPM mechanism based on iteratively rearranging the dimension in a given zero-one data space according to the scoring (banding) mechanism until no more changes can be made thus avoiding permutation generations. Although BPM algorithms works well on reasonably sized N-D data, however large data that does not fit into primary storage still pose a challenge. Therefore, this paper thus proposea segmentation technique that identify banding in large data sets. Using this technique, we partition(segment) data into manageable subsets; identify local banding on each partitions and then we apply the best local banding configuration on the entire dataset to obtain a global banding. The proposed segmentation (partition) technique however, considers a reference dimension resulting in individual locations within the remaining (N- 1) D datasets to holds more than one entries per location.

MATERIALS AND METHODS

A number of researcher have investigated the zero-one banding; the notable algorithms include Minimum Banded Augmentation (MBA) (Gemma et al, 2008, Gemma et al, 2011), that assumed a fixed column permutation; the algorithm considers series of column permutations to produce a number of permuted matrices (ordered matrices).Another algorithm is the Barycentric (BC) (Makinen and Siirtola, 2005);based on barycentre measure to identify

We define data space in terms of (l_1,l_2,l_3, …..,l_n) hyper-grid, where l_1 represents dimension one size (dim_1), l_2 represents the dimension two size(dim_2). It is worth noting that the dimensions are not of equal size. For N-D, the set of dimensions Dim = {dim_1,dim_2, …,dim_n}. Thus, each non-zero entries in theN-D space is represented by set of coordinates (indexes) {c_1, c_2, ….,c_n} (where n is the size dimensions) and $c_1 \in dim_1$, $c_2 \in dim_2$, ... , $c_n \in dim_n$.

The approximate BPM operates by considering possible dimension pairings: **Max pairings = |Dim| × (|Dim|-1)**. For each pairing, a banding score for index x in Dim_i calculated with respect to $Dim_j$and multiply by the number of dots using Equation 1.The calculated banding score values are used to arrange indexes in Dim_i

permutation of rows/columns, BC operates by calculating the average index locations of one entries within each row (column). The permutations generation was an NP-complete problem, as such the algorithms in (Gemma et al, 2008 and Makinen and Siirtola, 2005), were only restricted to 2D data. Banded Pattern Mining (BPM) idea was proposed first in (Abdullahi et al, 2014b;Abdullahi et al, 2015a; Abdullahi et al, 2015b; Abdullahi et al, 2016 and Abdullahi et al, 2018). The paper thus identify bandings in N-Dimensional (N-D) data set susing BPM algorithm that incorporate banding score mechanism. The BPM algorithm operate by calculate banding scores for individual indexes in the dimensions and reorder the dimension indexes in ascending order. The alternative versions of the algorithms presented were the; Approximate version and Exactversion, conceived from the preliminary work presented in (Abdullahi et al, 2014b;Abdullahi et al, 2015a; Abdullahi et al, 2015b; Abdullahi et al, 2016a; Abdullahi et al, 2016b and Abdullahi et al, 2018). The quality of the overall banding scores for the given dataset measured by calculating the individual scores and normalising to produce an Overall Banding Score (OBS). From the perspective of data analysis, the concept of banded matrices is significant and cut across many applications domains: (i) network analysis: represents the relationship between data entities. (Gemma et al, 2008), (ii) co-occurrence analysis: identify two different categories of entity that co-occur. (Manilla et al, 2006), (iii)VLSI chip design: the process of creating an integrated circuit by combining thousands of transistors into a single chip (Koebe and Knochel, 1990).

Banding Score Calculation

and subsequently the data space. The overall banding score is calculated using Equation 2 by summingthe entire BS and dividing by the dimension size (|Dim|).

$$BS_{ijx} = \frac{\sum_{x=1}^{x=|N_{ijx}|} n_x \times m_x}{\sum_{y=1}^{y=|N_{ijx}|} \left(Dim_y - k + 1\right) \times m'_y} \quad (1)$$

Where: (i) $N_{ijp}$is the set of indexes location that features index p in Dimi and hold one of multiple dots. N={n1,n2,….}. m_x is the number of dots at location $n_x m_x \in M_{ijp}$, $M_{ijp}$is the number of dots in index p with respect to Dimj. M_{ijp}= {m1,m2,…..}, $|M_{ijp}| = |N_{ijp}|$. The $m'_y$is the set of location quantities M_{ijp} in descending order of the size of element. Dim is the set of dimension.

$$OBS = \frac{\sum_{i=1}^{i=|Dim|}\sum_{j=1}^{j=|Dim|^{j\neq i}} \frac{\sum_{x=1}^{x=Dim_i} BS_{ijx}}{Dim_i}}{|Dim|\times|Dim|-1} \quad (2)$$

The exact BPM operates by considering the whole data space; the banding score for each index p in Dim_i calculated using Equation 3. The banding score value is used to rearrange the indexes in Dim_i to reconfigure D. The overall banding score (OBS) is calculated using Equation 4, by summing the entire individual BS and divided by the number of dimension. The exact algorithm requires a set of S maximum distance calculations, where S is the number of dots under considerations. Equation 3 calculates the Banding score.

$$BS_{ix} = \frac{\sum_{x=1}^{x=|N|} dist(n_x \in N)\times m_y \in M}{\sum_{y=1}^{y=|W|}(w_y \in W)\times m_y \in M'} \quad (3)$$

OBS calculated by adding up and normalising the entire individual BS associated with individual dimensions, shown in Equation 4:

$$OBS = \frac{\sum_{i=1}^{i=|Dim|} \frac{\sum_{x=1}^{x=Dim_i} BS_{ix}}{Dim_i}}{|Dim|} \quad (4)$$

Banded Pattern Mining Mechanism

The banding score concept is incorporated into the BPM algorithms. The banded data set is expressed as the Overall Banding Score (OBS), a number from 0 to 1. An OBS zero (0)will be achieved when the whole data space is filled with zero entries and OBS one (1) when the whole data space is filled with one entries (1). However, it is worth noting that the best OBS is zero and the worse is one. The BPM approximate and BPM exact algorithm works in a similar manner at a high level. The algorithms iteratively loops, on each iteration:

1. It rearranges indexes in the Dim using the BS to produce a reconfigured Dim^1, Equation 1 &3.
2. The revised set of dimension Dim^1 used to rearrange the data space D to give D^1.
3. A new OBS values calculated using Equation 3 and 4.
4. If the new OBS value is greater than or equal to the current OBS (OBS). The algorithm exit with the current configuration and OBS.
5. Otherwise, D, Dim and OBS values are updated and the algorithm repeat.

BPM Segmentation

The BPM approximate and BPM exact algorithms both identify bandings in data. However, our interest is on large N-D data, which are too large to be contained in the primary storage of a single machine. The idea is the adoption of a segmentation technique. The idea is to identify banding sequentially using a sequence of data segments R taken from a large N-D data set and then combine the different bandings on completion. The advantage of using segmentation technique is that, the entire data set will be taken into account. In the evaluation presented, the size of R thus depends on the size of the data subset that can be processed at any one time. In this paper, the size of |R| was three. The problem with this approach is how to aggregate the results to produce the best overall configuration. The technique used in this paper, conducts bandings on sequence of data segments, by selecting the best OBS banding and applying to the entire data set.

Worked Example

Figure 2, presents a worked example showing the operation of the BPM algorithm. A 4x4 2D configuration with multiple dots in cells.

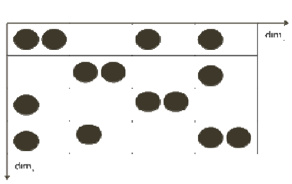


Figure 2: 2D Matrix

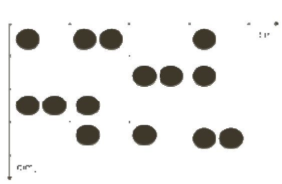
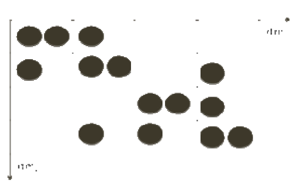


Figure 3:Dim_xrearranged



Figuure4: Dim_y rearranged

Figure 2: 2D Banded matrix (a) 2D matrix, (b)2D matrix after Dim_x(column) rearranged and (c) 2D matrix after Dim_y (row) rearranged

The matrix features: Dim = {x:y}, where Dim_x comprise of {0, 1, 2, 3} and Dim_y comprise of {0, 1, 2, 3} with multiple dots in some locations. The BPM algorithm commence with Dimx. By calculating banding scores taking into consideration the number of non-zero entriesper location, this is presented in Table I. The algorithm then arrange the indexes in Dim_x according to the banding score in ascending order to obtain Figure 3.

Table I: Banding score results forDim_x (First iteration)

Index	**Distance from Origin**	**Maximum distance From Origin**	**BS**
0	(0 * 2) + (2 *1)+ (3 * 1) = 5	(1 * 1) + (2 *1)+(3 * 2) = 9	0.556
1	(1 * 2) + (3 * 1) = 5	(2 *1) +(3* 2) = 8	0.625
2	(0 *1) + (2 * 2) = 4	(2 *1) + (3 * 2) = 8	0.500
3	(0 *1) + (1 *1) +(3*2) = 7	(1 * 1) + (2 * 1)+(3 *2) = 9	0.778

Next, Table II presents the banding scorescalculation for Dim_y taking into consideration the number of non-zero entries per location). The algorithm then reorder the elements in Dim_y in ascending order to obtain the configuration shown in Figure 4:

Table II:Banding score results forDim_y (second iteration)

Index	**Distance from Origin**	**Maximum distance From Origin**	**BS**
0	(0 * 1) + (1 *2)+(3*1) = 5	(1*1)+(2 *1)+(3 * 2) = 9	0.556
1	(2 * 2) + (3*1) = 7	(2 *1) + (3*2) = 8	0.875
2	(0 *2) + (1 * 1)= 1	(2 *1) + (3 *2)= 8	0.125
3	(1 *1) + (2 *1)+(3*2) = 9	(1 * 1) + (2 * 1) +(3 * 2) = 9	1.000

The calculation of OBS using Equation 2:

$$OBS = \frac{5.0}{9.0} + \frac{5.0}{8.0} + \frac{4.0}{8.0} + \frac{7.0}{9.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{1.0}{8.0} + \frac{9.0}{9.0} = 0.6324$$

The algorithm then test the new OBS value against the previous value. If (OBS_{new}< OBS) the new OBS becomes the OBS (OBS = OBS_{new} and D = D′). The algorithm then repeat the process until the maximum number of iterations reached. The same banding scores for Dim_x and Dim_y were obtained on the third iteration as shown in Table III.

Table III: Banding score results for Dim_x and Dim_y(Third iteration)

Index	**Distance from Origin**	**Maximum distance from Origin**	**BS**
0	(0 * 2) + (1*1)= 1	(2 *1)+(3 * 2) = 8	0.125
1	(0 * 1) + (1 *2) + (3*1)= 5	(1*1)+(2 *1) +(3*2) = 9	0.556
2	(2*2) + (3 *1)= 7	(2 *1) + (3 * 2)= 8	0.875
3	(1 *1) + (2 * 1)+(3*2) = 9	(1 *1) + (2 * 1)+(3 *2) = 9	1.000

The OBS is

$$\frac{1.0}{8.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{9.0}{9.0} + \frac{1.0}{8.0} + \frac{5.0}{9.0} + \frac{7.0}{8.0} + \frac{9.0}{9.0} = 0.6471$$

RESULTS AND DISCUSSION

The evaluation and the discussion of the proposed segmentation compatible with BPM and BPM algorithms is presented in this section. The experimental analysis presents the result for4-D and 5-D data sets extracted from the Great Britain (GB) Cattle Tracking System (CTS). The four dimensions (4-D) comprises of: (i) records, (ii) attributes, (iii) sender easting and (iv) sender northing and the five dimensions features: (i) records, (ii) attributes, (iii) sender easting, (iv) sender northing and (v) time. The data sets from the years2003 to 2004 was consider with respect to Aberdeenshire and Cornwall counties. The Eastings and Northings data sets discretised into ten sub-ranges, whilst the temporal dimension into 12 sub-ranges each describing a month. The attribute dimension comprised the following individual attributes: (i) animal gender, (ii) animal age, (iii) cattle-beef, (iv) cattle-dairy, (v) sender location type, (vi) receiver location type and (vii) the number of cattle moved. The attribute values were also discretised/normalised using the LUCS-KDD DN Software (Coenen, 2003). In each case, data set were partitioned into a sequence of data segments, each segment represents 4 months in a particular year, thus 3 segments per data set. The evaluation objectives are:

1. To compare the effectiveness of BPM approximate and BPM exact algorithms using Overall Banding Score (OBS).
2. To compare the efficiency of the BPM approximate and BPMexact algorithms computational time (in seconds).

Comparison of BPM Approximate and BPM exact

This section presents the experimental reports conducted to determine effectiveness of BPM algorithms in the context OBS values. This evaluation consider 4-D and 5-D data sets taken from the CTS database. Each data set were grouped into a sequence of three data segments, three because this corresponded to the maximum number of data that can be easily processed on a single machine. Tables VI give the result for the effectiveness of banding obtained expressed in terms of OBS values a number between zeros and ones, where OBS value 0 indicate the best results highlighted in colour font. The table presents OBS values to: (i) segment data, (ii) original data set with best segment banding configuration and (iii) original data set without banding. The columns represents the two counties (Aberdeenshire and Cornwall), the year and the OBS value. From Table VI, There was an improvement on the OGB value from the original data set after applying the result from best-selected segment banding configuration. However, the segment bandings improved over the overall banding (this is to be expected). The paper also show that the BPM approximate and BPM exact mechanism work well in large N-D data set with reasonable processing time. In addition, the use of the proposed segmentation technique had shown to produce effective bandings in the terms of OBS value than when there is no banding on the datasets. In addition, the BPM exact produced the best OBS value.

Table VI: Aberdeenshire and Cornwall OBS results (best results highlighted in colour font)

Counties	Data sets	Years	OBS			
			BPMexact		BPM approximate	
			4D	5D	4D	5D
	Banding of Segment 1	2003	0.2438	0.1802	0.3471	0.3374
	Banding of Segment 2		**0.238**	**0.174**	0.3450	0.3281
	Banding of Segment 3		**6**	**1**	0.3906	0.3753
	Final band. Segment (best OBS)		0.2859	0.2066	0.4040	0.4241
Aberdeen-	No banding		**0.282**	**0.234**	0.4242	0.4415
shire			**4**	**3**		
			0.2924	0.2383		
	Banding of Segment 1	2004	0.2622	0.1729	0.3754	0.3707
	Banding of Segment 2		0.2458	0.1818	0.3537	0.3445
	Banding of Segment 3		**0.240**	**0.176**	0.3463	0.3293
	Final band. Segment (best OBS)		**9**	**1**	0.3502	0.3596
	No banding		**0.245**	**0.205**	0.4595	0.4837
			6	**8**		
			0.3290	0.2671		
	Banding of Segment 1	2003	**0.254**	**0.185**	0.3654	0.3490
	Banding of Segment 2		**1**	**3**	0.3775	0.3533
Cornwall	Banding of Segment 3		0.2607	0.1875	0.4078	0.3832
	Final band. Segment (best OBS)		0.2795	0.2053	0.4075	0.3803
	No banding		**0.284**	**0.201**	0.4570	0.4831
			4	**5**		
			0.3213	0.2577		
	Banding of Segment 1	2004	**0.236**	0.2093	0.3979	0.3916
	Banding of Segment 2		**2**	0.2086	0.4017	0.3870
	Banding of Segment 3		0.2420	**0.203**	0.3949	0.3736
	Final band. Segment (best OBS)		0.2448	**6**	0.4214	0.4236
	No banding		**0.257**	**0.233**	0.4633	0.4821
			9	**6**		
			0.3281	0.2657		

Tables VII: Aberdeenshire and Cornwall run time results (best results highlighted in colour)

Counties	Data sets	Years	Computational time(in seconds) BPM exact 4D	5D	BPMapproximate 4D	5D
	Banding of Segment 1	2003	18.53	33.82	**14.04**	**29.30**
	Banding of Segment 2		19.96	32.04	**15.97**	**26.54**
Aberdeen-shire	Banding of Segment 3		19.67	36.50	**15.35**	**27.41**
	Final band. Segment (best OBS)		16.38	31.42	**13.88**	**26.36**
	Banding of Segment 1	2004	18.83	33.06	**14.70**	**26.13**
	Banding of Segment 2		18.70	32.49	**14.37**	**28.36**
	Banding of Segment 3		19.96	32.49	**15.12**	**28.15**
	Final band. Segment (best OBS)		16.98	32.80	**12.04**	**26.10**
	Banding of Segment 4	2003	19.30	36.29	**13.23**	**28.30**
	Banding of Segment 5		17.17	36.58	**14.10**	**27.08**
Cornwall	Banding of Segment 6		16.40	35.50	**13.07**	**27.51**
	Final band. Segment (best OBS)		15.84	34.81	**14.86**	**27.20**
	Banding of Segment 1	2004	20.32	36.20	**12.19**	**28.24**
	Banding of Segment 2		18.13	35.46	**13.44**	**27.61**
	Banding of Segment 3		20.87	35.18	**14.13**	**28.54**
	Final band. Segment (best OBS)		16.82	34.72	**11.38**	**28.07**

Table VII presents the runtime results obtained in seconds, with best results highlighted in colour font. The table presents the results for Aberdeen and Cornwall counties. Note that the runtime results presented in the tables are for: (i) banding run time with respect to each segments, (ii) the final banding run time on the original data. From the tables, it can be seen that BPM approximate was computationally more efficient than BPM exact. Because it require less processing time. To further, understand the results from Table VI. Figure 5 and 6 present the bar graph representation of the results. From the figures, an improvement can be seen on the original data after applying the best banding configuration on the original datasets than original data without banding.

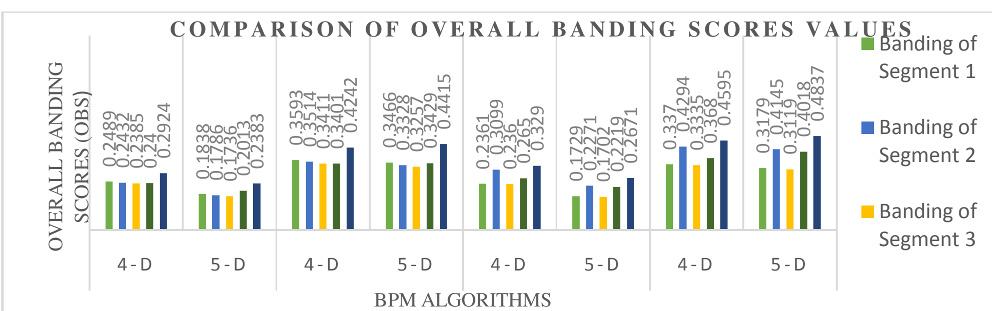


Figure 5: Comparisons of OBS values for Aberdeenshire county 5-D datasets using BPM approximate and BPM exact algorithms.

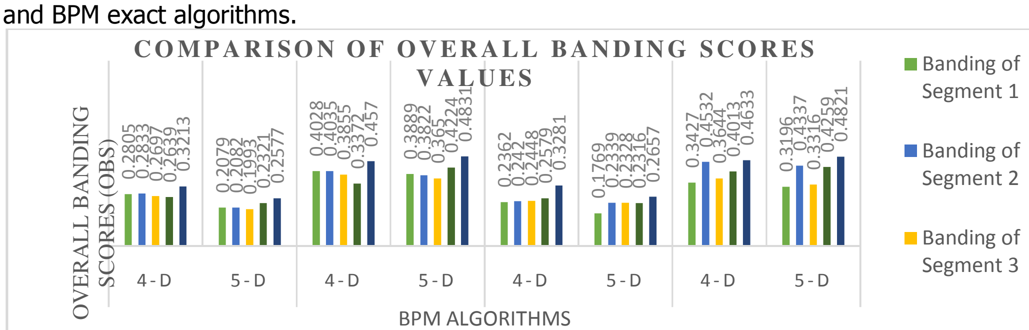


Figure 6: Comparisons of OBS values for Cornwall county 5-D datasets using BPM approximate and BPM exact algorithms.

CONCLUSION

This paper presents BPM algorithms that support large-scale data analysis. The paper address situations where we seek to establish sequentially bandings in large datasets using sequences of data segments. More specifically, the two banding algorithms considered were, BPM approximate and BPM exact. The evaluation presented established the following main findings. The BPM exact and BPM approximate mechanism were able to identify accurate bandings in large N-D data sets with reasonable processing time. The BPM approximate algorithm offered advantage of producing more efficient banding results in terms of runtime than the BPM exact algorithm; in addition, the quality of bandings produced from the BPM approximate algorithm were worse than those from BPM exact. Consequently, the most effective BPM algorithmin terms of Overall Banding Score (OBS) is the BPM exact. For future work, the author further intends to investigate banding in large N-D datasets using both sampling and segmentation techniques. Whatever the case, the author have been greatly encouraged by the results obtained as presented in this paper.

REFERENCES

Abdullahi, F.B., Coenen, F., & Martin, R. (2014). A novel approach for Identifying banded patterns in zero-onedata using column and row banding scores," Proceedings of International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, 2014a, pp. 58–72.

Abdullahi, F.B., Coenen, F., & Martin, R. (2015). Finding banded patterns in bigdata using sampling," in 2015 IEEE International Conference on Big Data (Big Data).IEEE, 2015a, pp. 2233–2242.

Abdullahi, F.B., Coenen, F., & Martin, R. (2014). A scalable algorithm forbanded pattern mining in multidimensional zero-one data, Proceedings of International Conference onDataWarehousing and Knowledge Discovery. Springer, 2014b, pp. 345–356.

Abdullahi, F.B., Coenen, F., & Martin, R. (2015). Finding banded pattern in data: The banded pattern mining algorithm, Proceedings of International Conference on Big DataAnalyticsand Knowledge Discovery. Springer, 2015b, pp. 95–107.

Abdullahi, F.B., Coenen, F., & Martin, R. (2016). Banded pattern mining Algorithms in multi-dimensionalzero-one data," in Transactions on Large-Scale Data andKnowledge-Cantered Systems XXVI. Springer, 2016a, pp. 1–31.

Abdullahi, F.B. (2016). Banded Pattern Mining for N-Dimensional Zero-One data PhD Thesis, University of Liverpool, United Kingdom, 2016b.

Abdullahi, F.B., & Coenen, F. (2018)**.** Multi-Dimensional Banded Pattern Mining in Proceedingsof 15th Pacific Rim Knowledge Acquisition Workshop (PKAW2018),Springer LNAI 11016, pages. 154–169, 2018.

Cheng,K. Y. (1973). Minimising the bandwidth of sparse symmetric matrices, in Computing, 1973, vol. 11, pp. 103–110.

Cuthill, A. E. & McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices." in Proceedings of the 24th National Conference of ACM, 1969, pp. 157–172.

Fortelius, M., Kai Puolamaki, M.F., Manilla, H. 2006) Seriation in paleontological data using Markov chain monte carlo methods. PLoS Comput. Biol. **2**, 2 (2006)

Gemma, G .C, Juntilla, E., & Manilla, H. (2008). Banded structures inbinarymatrices," in Proceedings Knowledge Discovery in Data Mining(KDD08), 2008, pp.292–300.

Gemma, G .C, Juntilla, E., &Manilla, H. (2011).Banded structure inBinarymatrices,
Knowledge and Information Systems, vol. 28, no. 1,pp. 197–226,2011.

Juntilla, E. (2011). Pattern in Permuted Binary Matrices. PhD thesis, 2011.

Koebe, M., & Knochel, J. (1990). On the block alignment problem, Elektronisch Informationsverarbeitung und Kybernetik, vol. 26, No. 7,pp. 377–387, 1990.

Makinen, E and Siirtola, H. (2005). The barycenter heuristic and thereorderable matrix, Informatica, vol. 29, pp. 357–363, 2005.

Papadimitriou,C. H. (1976). The np-completeness of the bandwidth minimisation problem, in Computing, vol. 16, 1976, pp. 263–270.

Mannila,H and Terzi .E, (2007). Nested and segmented nestedness. In: Proceedings of the 13th ACMSIGKDD international conference on knowledge discovery and data mining, New York, NY, USA, 2007. pp. 480-489.

COENEN, F., (2003), LUCS-KDD Data generator Software. Department of Computer
Science, the University of Liverpool UK. http://www.liv.ac.uk/_frans/KDD/Software/LUCS_KDD DataGen Generator.html,