A Novel Approach For Identifying Banded Patterns In Zero-One Data Using Column and Row Banding Scores

Fatimah B. Abdullahi, Frans Coenen, and Russell Martin

The Department of Computer Science, The University of Liverpool, Ashton Street, Liverpool, L69 3BX, United Kingdom
email: {f.b.abdullahi,coenen,russell.martin}@liverpool.ac.uk

Abstract. Zero-one data is frequently encountered in the field of data mining. A banded pattern in zero-one data is one where the attributes (columns) and records (rows) are organized in such a way that the "ones" are arranged along the leading diagonal. The significance is that rearranging zero-one data so as to feature bandedness enhances the operation of some data mining algorithms that work with zero-one data. The fact that a dataset features banding may also be of interest in its own right with respect to various application domains. In this paper an effective banding algorithm is presented designed to reveal banding in 2D data by rearranging the ordering of columns and rows. The challenge is the large number of potential row and column permutations. To address this issue a column and row scoring mechanism is proposed that allows columns and rows to be ordered so as to reveal bandedness without the need to consider large numbers of permutations. This mechanism has been incorporated into the Banded Pattern Mining (BPM) algorithm proposed in this paper. The operation of BPM is fully discussed. A Complete evaluation of the BPM algorithm is also presented clearly indicating the advantages offered by BPM with respect to a number of competitor algorithms in the context of a collection of UCI Datasets.

Keywords: Banded Patterns, Zero-One data, Data Mining and Pattern Mining

1 Introduction

Many data sets take the form of a $n \times m$ two dimensional, binary ("zero-one"), matrix whereby a one indicates the presence of some attribute and a zero its absence with respect to a particular record. More formally if we consider a data set of attributes $A = \{a_1, a_2, \ldots, a_m\}$ with the value set $\{0, 1\}$, and a set of records $R = \{r_1, r_2, \ldots, r_n\}$ such that each record is some subset of A, then we have a binary valued, zero-one, data set. Many application domains exist where zero-one data is typically found, examples include: the field of information retrieval [5], bioinformatics and computational biology (genes, probe mappings) [3],[18] and paleontology (sites and species occurrences) [4], [10].

Of note with respect to this paper is the field of data mining, especially frequent pattern mining [1, 2, 8] where it is necessary to process large collections of zero-one data stored in the form of a set of feature vectors (drawn from a vector space model of the data). Although outside the scope of this paper, banding will also have benefits with respect to non-binary data. For example in the context of co-clustering algorithms where the objective is to simultaneously cluster the rows and columns in 2D data set (see for example the work presented in [12] and [16].

A zero-one matrix is fully banded if both the columns and rows can be presented in such a manner that the "ones" are arranged along the leading diagonal as illustrated in Figure 1. In practice data can typically not be perfectly banded, but in many cases some form of banding can be achieved.

Fig. 1. Banding Example: (a) raw data (b) rearrangement of columns and rows to reveal a banding

While the concept of banded matrices has its origins in numerical analysis, it has been studied within the data mining community. The benefits of banding may be summarized as follows:

1. The fact that a zero-one data set can be banded is interesting in its own right as it is indicative of the existence of a relationship between the columns and rows, for example between records and certain attribute values (assuming rows equate to records and columns to attribute values).
2. Working with banded data is seen as preferable from a computational point of view as the computational cost involved in performing certain operations falls significantly for banded matrices, often leading to significant savings in terms of processing time [9]. One example where this is the case is the use of affinity matrices in spectral clustering [15], another example is where adjacency matrices are used in the context of graph mining [13]. A third example is in the context of co-clustering algorithms [12] and [16].
3. Related to 2 above, working with banded patterns requires less storage. When a matrix is banded only the non-zero diagonal need to be stored. Thus, for the banded storage schemes the amount of memory required to store the matrix is directly proportional to the bandwidth. Therefore, finding a row-column ordering that minimizes the bandwidth is significant with respect to reducing storage space and consequently results in algorithmic speed up [17].

The main issue with the identification of banding in data is the large number of permutations that need to be considered. In this paper we present an approach whereby bandedness in data can be revealed using a scoring mechanism thus avoiding the need to consider permutations. This concept has been built into the Banded Pattern Mining (BPM) algorithm; the central contribution of this paper. This differs Minimum Banding Augmentation (MBA) algorithm proposed in [14], see above, which allowed for the discovery of banded structures in binary matrices by assuming a fixed column permutation.

The rest of this paper is organized as follows. Section 2 disuss related work. A formalism for the banded pattern problem is then presented in Section 3. This is followed, in Section 4, with an overview of the proposed scoring mechanism. Section 5 presents the BPM Algorithm whilst Section 6 provides a worked example illustrating the proposed approach. The evaluation of the BPM algorithms is then reported in Section 7; and finally, in Section 8, some conclusions are presented.

2 Related Work

The property of bandedness with respect to data analysis was first studied by Gemma *et al.* [11]. Where the minimum banding problem was addressed by computing "how far" a 2-D data matrix (data set) was away from being banded. The authors in [11] define the banding problem as: given a binary matrix M, find the minimum number of 0s entries that needs to be modified into 1s entries and the minimum number of 1s entries that needs to be modified into 0s entries so that M becomes fully banded. In [11], the authors use the principle of assuming "a fixed column permutation" on a given Matrix M. The basic idea is to solve optimally the consecutive one property on the permuted matrix M and then resolve Sperner conflicts between each row of the permuted matrix M, by going through all the extra rows and making them consecutives. While it can be argued that the fixed column permutation assumption is not a very realistic assumption with respect to many real world situations, heuristical methods were proposed in [11] to determine a suitable fixed column permutation.

The current state of the art algorithm is the MBA algorithm [14] which also adopts the fixed column permutation assumption. The MBA algorithm focuses on minimizing the distance of non-zero entries from the main diagonal of the matrix by reordering the original matrix. The MBA algorithm operates by "flipping zero entries (0s) into one entries (1s) and vice versa to identify a banding.

Another Strategy for transposing a zero-one matrix is the Barycentric (BC) algorithm that was previously used to draw graphs [20] to seriate paleontological data [7], and more recently used to reorder binary matrices [19]. In essence, the Barycentric algorithm finds permutations for both rows and columns such that 1*s* are as close to each other as possible. It is based on the Barycentric measure, which is the average position of 1s in a row/column. The algorithm first computes the barycenter for all rows, then orders the rows from smallest to largest,

transposes the matrix accordingly and then iterates again until convergence is reached.

Given the above the MBA and BC algorithms are the two exemplar banding algorithms with which the operation of the proposed BPM algorithm was compared and evaluated as discussed later in this paper (see Section 7).

3 Problem Definition

A binary matrix is a matrix with entries $\{0, 1\}$. Let A be an $n \times m$ binary matrix comprising rows $\{1, 2, \ldots, n\}$ and columns $\{1, 2, \ldots, m\}$. We indicate a particular row i using row_i $(1 \leq i \leq n)$, a particular column j using col_j $(1 \leq j \leq m)$ and a particular element using a_{ij}. We denote the set of '1' valued elements associated with row i using R_i, $R_i = \{r | \forall a_{ij} \in row_i, a_{ij} = 1\}$. Similarly the set of '1' valued elements associated with column j using C_j, $C_j = \{c | \forall a_{ij} \in col_j, a_{ij} = 1\}$. (It may ease understanding to note that C_j is sometimes referred to as a Transaction ID list or a TID list, a concept well established in transaction rule mining [1, 2]). A zero-one matrix A can be "perfectly banded" if there exist a permutation of columns $\{1, 2, \ldots, m\}$ and rows $\{1, 2, \ldots, n\}$ such that: (i) for every element in C_j the 1 values occur consecutively at row indexes $\{i_k, i_{k+1}, i_{k+2}, \ldots\}$ and the "starting index" for C_j is less than or equal to the starting index for C_{j+1}; and (ii) for every element in R_i the 1 values occur consecutively at column indexes $\{j_k, j_{k+1}, j_{k+2}, \ldots\}$ and the starting index for R_i is less than or equal to the starting index for R_{i+1}.

4 The Banding Score Mechanisms

The discovery of the presence of banding in zero-one matrices requires the rearrangement of the columns and rows in the matrix so as to "reveal" a banding (or at least an approximate banding if no perfect banding exists). As noted above the discovery of banded patterns in zero-one data requires some mechanism for automatically determining whether one potential configuration features a "better" banding (according to the above definition) than some other configuration. The idea presented in this paper is to use the concept of "banding scores" that reflect the relative position of each "one" value both horizontally and vertically. Consider the perfect banding given in Figure 2(a) where we have three records and an attribute set $\{a, b, c\}$, each record features one of the attributes. We wish to assign some kind of banding score to this configuration. We achieve this by considering two types of banding score: (i) *column scores* (BS_{col}) and (ii) *row scores* (BS_{row}). Both are determined in a similar manner thus the following discussion will be focused on column score calculation.

With reference to Figure 2(a), which features a perfect banding, we wish to assign a greater weight to locations corresponding to the first column than locations associated with the second column, and greater weight to locations associated with the second column than the third column (etc). In real datasets a perfect banding may not exist. However, we are interested in revealing the

Fig. 2. Example zero-one data configurations: (a) perfect banding, and (b) an alternative banding

arrangement that is as close to a perfect banding as possible. Thus for each column i we can calculate a Column Score CS as follows (we assume that column and row numberings start from 1):

$$CS = \sum_{k=1}^{k=|C_i|} m - r_k + 1$$

where m is the number of rows, C_i is the TID list for column i, and r_k is the row index at location k in C_i. Thus the column scores for the configuration in Figure 2(a) will be:

$$CS_A = 3-1+1 = 3 \qquad CS_B = 3-2+1 = 2 \qquad CS_C = 3-3+1 = 1$$

However, we would prefer it if our columns scores were normalized. Thus:

$$CS' = \frac{\sum_{k=1}^{k=|C_i|} m - r_k + 1}{\sum_{k=1}^{k=|C_i|} m - k + 1}$$

The normalised column scores will then be:

$$CS'_A = \frac{3-1+1}{3} = \frac{3}{3} = 1 \qquad CS'_B = \frac{3-2+1}{3} = \frac{2}{3} = 0.67$$

$$CS'_C = \frac{3-3+1}{3} = \frac{1}{3} = 0.33$$

We could now simply sum the individual column scores to obtain an overall column banding score. However, this would mean that the banding score for the configuration presented in Figure 2(a) would be equal to that for the configuration presented in Figure 2(b) (which features an entirely different kind of banding). Thus we need to weight the columns as well. Thus the final column banding score, BS_{col}, should be calculated as follows:

$$BS_{col} = \sum_{k=1}^{k=m} CS'_k(m-k+1)$$

As a result the configuration in Figure 2(a) will have a score of:

$$BS_{col} = 1 \times (3-1+1) + 0.67 \times (3-2+1) + 0.33 \times (3-3+1)$$

$$1 \times 3 + 0.67 \times 2 + 0.33 \times 1 = 4.67$$

The maximum normalized column score is 1, thus the maximum column banding score (BS_{col}), calculated as described above, will be 6. Consequently the normalized banding score should be calculated as follows:

$$BS'_{col} = \frac{\sum_{k=1}^{k=m} CS'_k(m-k+1)}{m(m+1)/2}$$

which means that the normalized column banding score for the configuration in Figure 2(a) is 0.78 while that for the configuration in Figure 2(b) is 0.56; thus the desired effect.

Considering the data set given in Figure 2 this can be configured in 6 ways. The different configurations are illustrated in Figure 3 together with their associated normalized column banding scores (BS'_{col}). From Figure 3, it can clearly be seen that the column banding score values serve to differentiate between the different configurations.

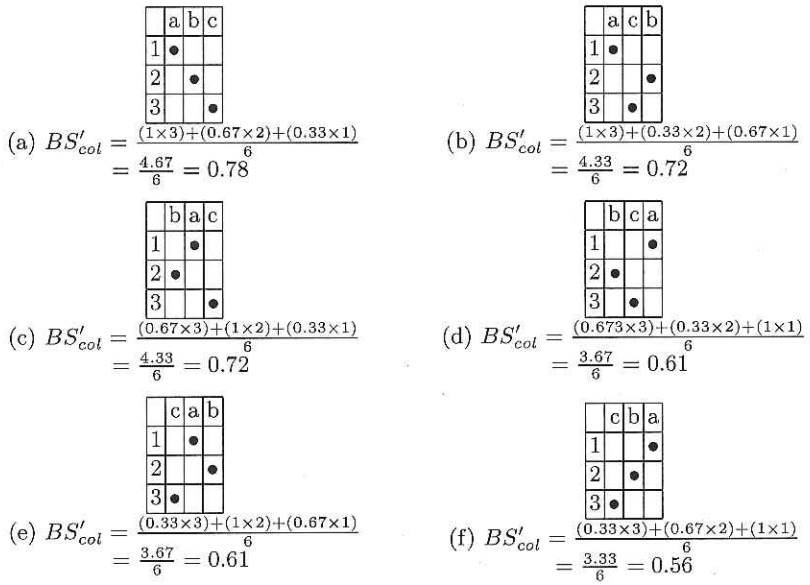

Fig. 3. Permutations for A = {a,b,c}

With respect to row banding scores (BS_{row}) we can formulate a similar argument. We can calculate individual Row Scores (RS) as follows:

$$RS = \sum_{k=1}^{k=|R_j|} n - c_k + 1$$

where n is the number of columns, R_j is the "column ID list" for row j, and c_k is the column index at location k in R_j. The normalized row scoes (RS') are then calculated as follows:

$$RS' = \frac{\sum_{k=1}^{k=|R_j|} n - c_k + 1}{\sum_{k=1}^{k=|R_j|} n - k + 1}$$

Thus the normalized row scores for the configuration presented in in Figure 2(a) will be:

$$RS'_a = \frac{3-1+1}{3} = \frac{3}{3} = 1 \qquad RS'_b = \frac{3-2+1}{3} = \frac{2}{3} = 0.67$$

$$RS'_c = \frac{3-3+1}{3} = \frac{1}{3} = 0.33$$

In this case the row and column scores are identical, this is because of the idealized banding featured in the configuration given in Figure 2(a).

The overall row banding score (BS_{row}) will then be given by:

$$BS_{row} = \sum_{k=1}^{k=n} RS'_k(n-k+1)$$

As a result the configuration in Figure 2(a) will have a score of:

$$BS_{row} = 1 \times (3-1+1) + 0.67 \times (3-2+1) + 0.33 \times (3-3+1)$$

$$1 \times 3 + 0.67 \times 2 + 0.33 \times 1 = 4.67$$

normalizing this:

$$BS'_{row} = \frac{\sum_{k=1}^{k=n} RS'_k(n-k+1)}{n(n+1)/2}$$

The overall normalized banding Score (BS) is then calculated as follows:

$$BS = \frac{BS'_{col} + BS'_{row}}{2}$$

The overall banding score for the configuration given in Figure 2(a) will then be 0.78 while that for the configuration given in Figure 2(b) will be 0.56.

5 Banded Pattern Mining Algorithm

From the above it was noted that to identify the "best" banding we need to maximize the banding score (BS). In this section the Banded Pattern Mining (BPM) algorithm is presented. The BPM algorithm was designed to identify a column and row configuration that serves to maximize BS. The algorithm proceeds in an iterative manner, on each iteration it sequentially rearranges the columns and rows according to their individual column and row scores (CS and RS). The process is continued until a maximum value for BS is reached. The BPM algorithm is presented in 1. The input is a zero-one matrix measuring $m \times n$ (line1).The algorithm proceeds in an iterative manner until BS is maximized or no changes have been made; initially BS is set to 0 (line 5). On each iteration the columns in A are first rearranged in descending order of BS'_{col} to produce a new matrix A' (lines 8-15), and then the rows in A' are rearranged in descending order of BS'_{row} to produce a new matrix A'' (lines 17-24). The normalized column and row banding scores are calculated (lines 16 and 25 respectively), and then a new banding score $newBS$ is determined (line 26). If $newBS$ is greater than the previously recorded normalized banding score we continue, if not we exit with matrix A as processed on the previous iteration. Thus if no changes (line 33) are made we also exit.

A disadvantage of the banding score calculation mechanism, as described above, might be that sparse data sets will generate very low banding scores as the normalization is conducted assuming a column and row score of one for every column and row. In some cases this might make it difficult to conduct comparisons depending on the precision of the numeric types used when implementing the above.

6 Worked Example

To illustrate the operation of the BPM algorithm, as described in the foregoing section, a worked example is presented in this section using the 5×5 input matrix shown in Figure 4. We commence, on the first iteration, by calculating the normalized column scores: $CS'_A = 0.7500$, $CS'_B = 0.7500$, $CS'_C = 0.8333$, $CS'_D = 0.5000$ and $CS'_E = 0.9167$. Using this set of scores the columns are rearranged to produce the matrix A' as shown in Figure 5. The normalized column banding score is now $BS'_{col} = 0.7444$. Next we calculate the normalized row scores: $RS'_1 = 1.0000$, $RS'_2 = 0.9167$, $RS'_3 = 0.7500$, $RS'_4 = 0.7500$ and $RS'_5 = 0.5000$. Using this set of scores the rows are rearranged to produce the matrix A'' as shown in Figure 6. The normalized row banding score is now $BS'_{row} = 0.8111$. The normalized banding score after this first iteration is then:

$$BS = \frac{BS'_{col} + BS'_{row}}{2} = \frac{0.7444 + 0.8111}{2} = 0.7776$$

On the second iteration we repeat the process. The normalized column banding scores are now: $CS'_E = 1.0000$, $CS'_C = 0.9167$, $CS'_A = 0.6667$, $CS'_B = 0.6667$

Algorithm 1 The BPM Algorithm

Input A, a zero-one matrix measuring $n \times m$
Output the matrix A rearranged so that the columns and rows serve to maximize BS
Column ordering = {1 ... m}
Row ordering = {1 ... n}
$BS \leftarrow 0$
loop
  $change \leftarrow false$
  CS' = empty set of column scores for columns 1 to m
  for all $j \in \{1 \ldots m\}$ **do**
    $CS'_j \leftarrow$ Normalized column score for column j
  end for
  $A' \leftarrow$ matrix A rearranged with columns in descending ordered as per CS'
  if $(A' \neq A)$ **then**
    $change \leftarrow true$
  end if
  $BS'_{col} \leftarrow$ Normalized column banding score
  RS' = empty set of row scores for row 1 to n
  for all $i \in \{1 \ldots n\}$ **do**
    $RS'_i \leftarrow$ Normalized row score for row i
  end for
  $A'' \leftarrow$ matrix A' rearranged with rows in descending ordered as per RS'
  if $(A'' \neq A')$ **then**
    $change \leftarrow true$
  end if
  $BS'_{row} \leftarrow$ Normalized row banding score
  $newBS' \leftarrow \frac{BS'_{col}+BS'_{row}}{2}$ {Normalized Banding Score}
  if $(newBS > BS)$ **then**
    $BS \leftarrow newBS$
    $A \leftarrow A''$
  else
    Exit with A
  end if
  if $(\neg change)$ **then**
    Exit with A
  end if
end loop

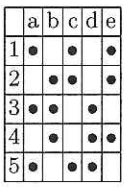

Fig. 4. Raw data

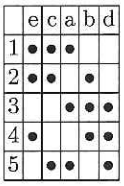

Fig. 5. Raw data with columns rearranged

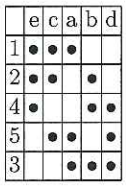

Fig. 6. Raw data with rows rearranged

and $CS'_D = 0.5000$; with this set of scores the columns remain unchanged.The overall normalized column banding score is $BS'_{col} = 0.8334$ (previously this was 0.7444). The normalized row banding scores are now: $RS'_1 = 1.0000$, $RS'_2 = 0.9167$, $RS'_4 = 0.6667$, $RS'_5 = 0.6667$ and $RS'_5 = 0.5000$. Again, with this set of scores the rows remain unchanged, thus produce the_same matrix A'' as was shown in Figure 6. The overall normalized row banding score is now $BS'_{row} = 0.8334$ (was 0.8111). The normalized banding score after this second iteration is now:

$$BS = \frac{BS'_{col} + BS'_{row}}{2} = \frac{0.8334 + 0.8334}{2} = 0.8334$$

On the previous iteration it was 0.7776, however no changes have been made on the second iteration so the algorithm terminates.

7 Evaluation

To evaluate the BPM algorithm, its operation was compared with the established MBA and BC algorithms, two exemplar algorithms illustrative of alternative approaches to identifying banding in zero-one data as described on Section 2. For the evaluation, eight data sets taken from the UCI machine learning data repository [6] were used, which featured nineteen columns (attributes) or more. The first set of experiments, reported in sub-section 7.1 below, considered the efficiency of the BPM algorithm in comparison with the MBA and BC algorithms. The second set of experiments (Section 7.2) considered the effectiveness of the BPM algorithm, again in comparison with the MBA and BC algorithms. The third set of experiments, reported in sub-section 7.3 below, considered the effectiveness of banding with respect to a frequent pattern mining scenario.

7.1 Efficiency

To determine the efficiency of the proposed BPM algorithm with respect to the MBA and BC algorithms, and with respect to the selected data sets, we recorded the number of iterations and run time required to maximize the banding score BS in each case. The data sets were normalized and discretized using the LUCS-KDD ARM DN Software[1] to produce the desired zero-one data sets (continuous

[1] http://www.csc.liv.ac.uk/~/frans/KDD/Software/LUCS_KDD_DN_ARM.

values were ranged using a maximum of five ranges). Table 1 shows the results obtained. The table presents the run-time and the final BS value obtained in each case. The table also records the number of columns (after discretization) and the number of rows. From the table it can be observed that there is a clear correlation between the number of rows in the dataset and run time. For example the largest dataset, Annealing, required considerably less processing time to identify a banding using BPM than when using either MBA and BC.

Fig. 7. Banding resulting from BPM algorithm as applied to Iris dataset (BS = 0.8404)

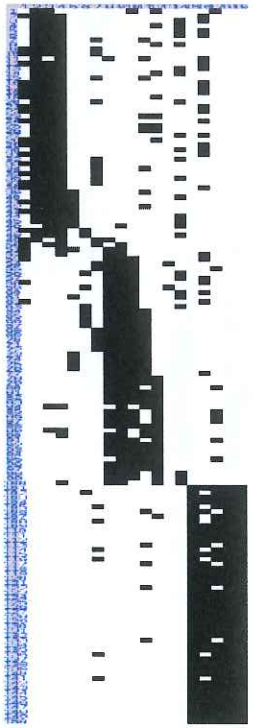

Fig. 8. Banding resulting from MBA algorithm as applied to Iris dataset (BS = 0.7806)

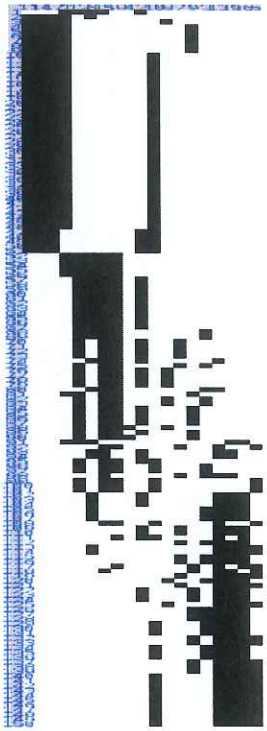

Fig. 9. Banding resulting from BC as applied to Iris dataset (BS = 0.7796)

7.2 Effectiveness With Respect to Banding Score

Note that, unsurprisingly, it was not possible to identify a perfect banding with respect to any of the UCI data sets. However, in terms of banding score, Table 1 clearly shows that the proposed BPM algorithm outperforms the previously proposed MBA and BC algorithms (best scores highlighted using bold font).

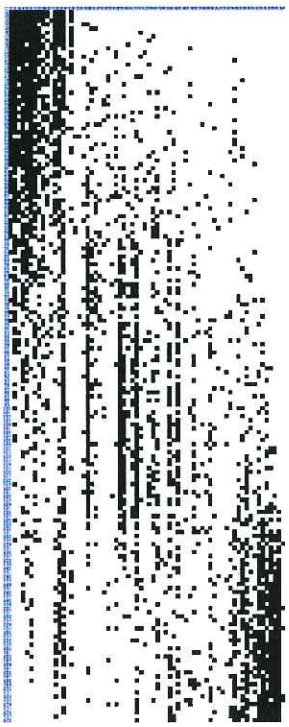

Fig. 10. Banding resulting from BPM algorithm as applied to Wine dataset (BS = 0.7993)

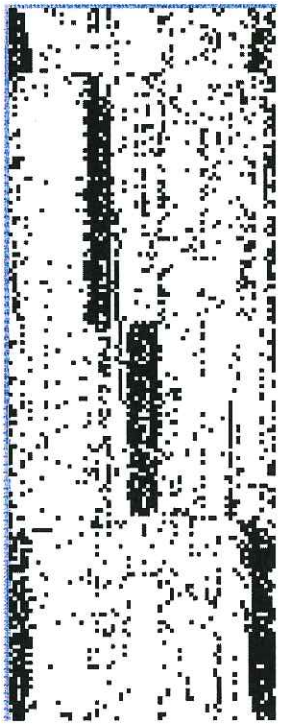

Fig. 11. Banding resulting from MBA algorithm as applied to Wine dataset (BS = 0.7123)

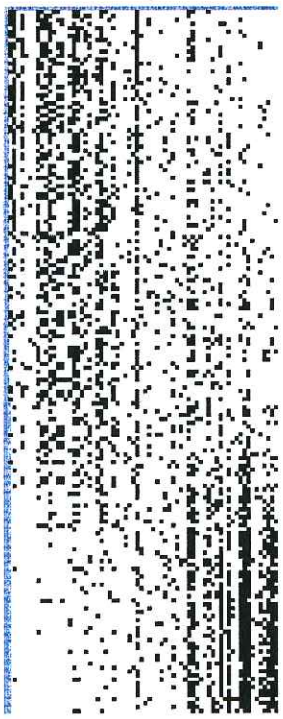

Fig. 12. Banding resulting from BC algorithm as applied to Wine dataset (BS = 0.7123)

Figures 7, 8, 9 and 10, 11, 12 show the bandings obtained using the Iris and Wine data sets using the BPM, MBA and BC algorithms respectively. Inspection of these Figures indicates that a banding can be identified in all cases. However, from inspection of the figures it is suggested that the bandings produced using the proposed BPM algorithm are better. For example considering the banding produced when the MBA algorithm is applied to the Wine dataset (Figure 11) the resulting banding includes "1"s in the top-right and bottom-left corners while the BPM algorithm does not (it features a smaller bandwidth). When the BC algorithm is applied to the Wine dataset (Figure 12) the banding is less dense than in the case of the BPM algorithm. Similar observations can be made with respect to the Iris data set.

7.3 Effectiveness With Respect to Frequent Pattern Mining

As already noted earlier in this paper banding has application with respect to increasing the efficiency of algorithms that use zero-one matrices or tabular in-

Table 1. Efficiency Experimental Results (best results presented in bold font), BS = Banding Score, RT = Run time (secs.)

Datasets	# Rec s	# Cols	BPM BS	MBA BS	BC BS	BPM RT	MBA RT	BC RT
annealing	898	73	**0.8026**	0.7305	0.7374	**0.150**	0.260	0.840
heart	303	52	**0.8062**	0.7785	0.7224	**0.050**	0.160	0.170
horsecolic	368	85	**0.8152**	0.6992	0.7425	**0.070**	0.200	0.250
lympography	148	59	**0.8365**	0.7439	0.7711	**0.030**	0.140	0.110
wine	178	68	**0.7993**	0.7123	0.7021	**0.040**	0.150	0.110
hepatitis	155	56	**0.8393**	0.7403	0.7545	**0.050**	0.150	0.090
iris	150	19	**0.8404**	0.8205	0.7516	**0.020**	0.080	0.060
zoo	101	42	**0.8634**	0.7806	0.7796	**0.020**	0.100	0.050

formation stored in the form of 2D data storage structures. One example is algorithms that use $N \times N$ affinity matrices, such as spectral clustering algorithms [15], to identify communities in networks (where N is the number of network nods). Another example is Frequent Itemset Mining (FIM) algorithms [1, 2] where it is necessary to process large collections of zero-one data stored in the form of a set of feature vectors (drawn from a vector space model of the data). To test the effectiveness of banding with respect to such algorithms a FIM algorithm was applied to the banded data sets produced using the proposed BPM and the established MBA algorithm as a result of the experiments reported in Sub-section 7.1 above. More specifically the Total From Partial (TFP) algorithm [8] was used, but any alternative FIM algorithm would equally well have sufficed. The results are presented in Tables 2 and 3 (a FIM support threshold, σ, of 2% was used). From the tables it can be seen that FIM is always much more efficient when using both BPM and MBA banded data than when using non banded data if we do not include the time to conduct the banding. If we include the banding time, in 8 out of the 12 cases for BPM and 4 out of the 12 cases for MBA it is still more efficient.

8 Conclusions

In this paper the authors have described an approach to identifying bandings in zero-one data using the concept of banding scores. The idea is to iteratively rearrange the columns and rows in a given zero-one matrix, according to normalized column (CS') and row (RS') scores until an overall banding score BS is maximized or no more changes can be made. These ideas have been incorporated into the Banded Pattern Mining (BPM) algorithm which was also described and illustrated. The BPM algorithm was evaluated by comparing its operation with the established MBA and BC banding algorithms using eight data sets taken from the UCI machine learning repository. The reported evaluation established that the proposed BPM approach could identify banding in zero-one data in a more effective manner (in terms of Banding score and run time) than in the

Table 2. FIM runtime with and without banding with BPM ($\sigma = 2\%$), best results highlighted in bold font

Datasets	#Rows	#Cols	Banding Time(s)	FIM time (s) with Banding	Total	FIM time (s) without Banding
adult	48842	97	346.740	**2.274**	349.014	**5.827**
anneal	898	73	0.150	**0.736**	**0.086**	2.889
chessKRvk	28056	58	95.370	**0.082**	95.452	**0.171**
heart	303	52	0.050	**0.294**	**0.344**	0.387
hepatitis	155	56	0.030	**0.055**	**0.085**	22.416
horseColic	368	85	0.070	**0.899**	**0.969**	1.242
letRecog	20000	106	42.420	**3.004**	45.424	**6.763**
lympography	148	59	0.030	**7.997**	**8.022**	12.658
mushroom	8124	90	14.400	**874.104**	**888.504**	1232.740
penDigits	10992	89	21.940	**2.107**	24.047	**2.725**
waveForm	5000	101	3.030	**119.220**	**122.250**	174.864
wine	178	68	0.010	**0.155**	**0.165**	0.169

Table 3. FIM runtime with and without banding with MBA ($\sigma = 2\%$)

Datasets	#Rows	#Cols	Banding Time(s)	FIM time (s) with Banding	Total	FIM time (s) without Banding
adult	48842	97	370.942	**10.525**	381.467	**5.827**
anneal	898	73	0.260	**1.733**	**1.993**	2.889
chessKRvk	28056	58	97.767	**0.075**	97.842	**0.171**
heart	303	52	0.160	**0.461**	0.621	**0.387**
hepatitis	155	56	0.150	**19.104**	**19.254**	22.416
horseColic	368	85	0.200	**2.134**	2.334	**1.242**
letRecog	20000	106	43.480	**6.221**	49.701	**6.763**
lympography	148	59	0.140	**11.187**	**11.329**	12.658
mushroom	8124	90	16.304	**1595.949**	1612.253	**1232.740**
penDigits	10992	89	22.002	**2.731**	24.733	**2.725**
waveForm	5000	101	3.135	**125.624**	**128.759**	174.864
wine	178	68	0.150	**0.211**	0.361	**0.169**

case of the MBA and BC comparator algorithms. The reported evaluation also confirmed that, at least in the context of FIM, efficiency gains can be realized using the banding concept. For future work the authors intend to extend their research on banding to address firstly banding in the context of 3D volumetric data, and then in the context of N dimensional data. The authors have been greatly encouraged by the results produced so far as presented in this paper.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD'93 pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings 20th International Conference on Very Large Data Bases (VLDB'94). pp. 487–499 (1994)

3. Alizadeh, F., Karp, R.M., Newberg, L.A., Weisser., D.K.: Physical mapping of chromosomes: A combinatorial problem in molecular biology. Algorithmica 13, 52–76 (1995)
4. Atkins, J., Boman, E., Hendrickson., B.: Spectral algorithm for seriation and the consecutive ones problem. SIAM J. Comput. 28, 297–310 (1999)
5. Baeza-Yates, R., RibeiroNeto., B.: Modern Information Retrieval. Addison-Wesley (1999)
6. Blake.C.I, C.J., M.: Uci repository of machine learning databases. http://www.ics.uci.edu/~mlearn/ ML Repository.htm (1998)
7. Brower, J.C., Kile, K.M.: Seriation of an original data matrix as applied to paleoecology. Lethaia 21, 79–93 (1988)
8. Coenen, F., Goulbourne, G., Leng, P.: Computing association rules using partial totals. In: Proceedings Data Mining and Knowledge Discovery (PKDD01), Spring Verlag LNAI 2168. pp. 54–66 (2001)
9. Cuthill, A.E., McKee, J.: Reducing bandwidth of sparse symmentric matrices. In: Proceedings of the 1969 29th ACM national Conference. pp. 157–172 (1969)
10. Fortelius, M., Puolamaki, M.F.K., Mannila, H.: Seriation in paleontological data using markov chain monte method. PLoS Computational Biology p. 2 (2006)
11. G.C, G., E, J., Mannila.H.: Banded structures in binary matrices. Knowledge Discovery and Information System 28, 197–226 (2011)
12. G.Pio, Ceci, M., D'Elia, D., Loglisci, C., Malerba., D.: A novel biclustering algorithm for the discovery of meaningful boiological correlaions between micrornas and their target genes. In: BMC Bioiinformatics. vol. 14, p. S8 (2013)
13. Inokuchi, A., Washio, T., H. Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Proceedings Principles of Knowledge Discovery in Databases (PKDD'00), Springer LNCS1910. pp. 13–23 (2000)
14. Junttila, E.: Pattern in Permuted Binary Matrices. Ph.D. thesis (2011)
15. Luxburg, U.V.: A tutorial on spectral clustering. Statistical Computation 17, 395–416 (2007)
16. M.Deodhar, Gupta, G., Ghosh, J., cho, H., Dhillon, I.: A scalable framework for discovering coherent co-clusters in noisy data. In: Proceedings of 26th International Conference on Machine Learning (ICML), Montreal Canada. p. 31 (2009)
17. Mueller, C.: Sparse matrix reordering algorithms for cluster identification. Machune Learning in Bioinformatics (2004)
18. Myllykangas, S., Himberg, J., Bohling, T., Nagy, B., Hollman, J., Knuutila, S.: DNA copy number amplification profiling of human neoplasms. Oncogene pp. 7324–7332 (2006)
19. Mkinen, E.., Siirtola, H.: The barycenter heuristic and the reorderable matrix. Informatica 29, 357–363 (2005)
20. Sugiyama.K, Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man and Cybernetics 11, 109–125 (1981)