Finding Banded Patterns in Big Data Using Sampling

Fatimah B Abdullahi
Department of Computer Science
University of Liverpool
Ashton Street Liverpool
L69 3BX United Kingdom
Email: f.b.abdullahi@liverpool.ac.uk

Frans Coenen
Department of Computer Science
University of Liverpool
Ashton Street Liverpool
L69 3BX United Kingdom
Email: coenen@liverpool.ac.uk

Russell Martin
Department of Computer Science
University of Liverpool
Ashton Street Liverpool
L69 3BX United Kingdom
Email: ramartin@liverpool.ac.uk

***Abstract*—A mechanism for identifying bandings in large "zero-one" N-dimensional data sets, using a sampling technique, is presented. The challenge of identifying bandings in data is the large number of potential permutations that need to be considered. To circumvent this a banding score mechanism is proposed that avoids the need to consider large numbers of permutations. This has been incorporated into a proposed banded pattern mining algorithm, the Exact ND Banded Pattern Mining (END BPM) algorithm. Although this operates well on reasonably sized datasets, there is still a challenge with respect to large N-dimensional data sets that cannot be held in primary storage. To this end a sampling technique is also proposed. The approach is fully described and evaluated using the GB cattle movement database, a "real life" database that records all movements of cattle in GB.**

***Keywords*—Banded Pattern in Big data, Banded Pattern Mining, Data Sampling**

I. Introduction

A binary valued data set is said to feature banding if the dimension indexes can be ordered in such a way that the "ones" are arranged about the leading diagonal [1], [2]. Typically, given a reasonably complex data set, a perfect banding can not be achieved, but some "best" banding is always possible. The advantage offered by banding is firstly that it allows for more efficient (and effective) processing of large data sets than in the case where the data sets are not organised in this manner; this is particularly significant in the case of very large data sets. Secondly it is often noteworthy that a banding can be identified in a given data set, as this tells us something of significance about the data. Examples of 2-Didemsional (2D) and 3-Dimensional (3D) bandings are presented in Figures 1 and 2.

The challenge of banding has traditionally been the large number of permutations to be considered. Existing work on identifying bandings in data has thus been mostly directed at 2D data and is typically founded on a generate and test process of dimension permutations [1]–[5]. A 2D binary data set can be viewed as a matrix holding 1s and 0s, if we ignore the 0s we are typically left with a sparse matrix (in this paper we consider the 1s in the matrix to be represented by "dots" and the 0s by "empty space"; thus we wish to arrange the dots to feature a banding). The problem of finding permutations of rows and columns in a given sparse matrix, such that the resulting matrix has a minimized bandwidth, is known to be NP-Complete [6]–[8]. To circumvent the need for this permutation generate and test process the central theme of this paper is the use of an ND banding score mechanism; a mechanism that includes the possibility of some cells in the matrix of interest holding more than one dot. The idea of using banding scores for Banded Pattern Mining (BPM) was first proposed in [9] where the authors proposed a 2D BPM mechanism such that the indexes in each dimension were allocated a banding score which could then be used to rearrange the indexes, thus avoiding the computationally expensive consideration of permutations. To the best knowledge of the authors very little work has been directed at ND data other than work on approximated bandings in 3D data described in [5]. However, it would be more desirable if an exact banding can be found. The first contribution of this paper is thus an Exact ND BPM (END BPM) algorithm that finds exact bandings in ND data which features the possibility that more than one dot may be located at individual locations in the data matrix. However, although this works well using reasonably sized ND data sets, large ND data sets that can not be held in primary storage still present a challenge; consequently the second contribution of this paper is a BPM sampling technique whereby large data sets can be approximately banded.

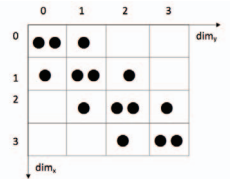


Fig. 1. $2D$ multiple dots configuration featuring a perfect banding

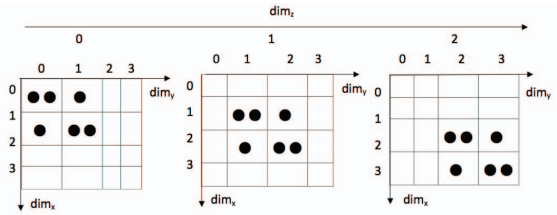


Fig. 2. $3D$ multiple dots configuration featuring a perfect banding

The concept of banded matrices, from the data analysis

perspective, is significant with respect to many application domains; examples can be found in paleontology [10], network data analysis [11] and linguistics [4]. The evaluation of the proposed END BPM algorithm considered in this paper was conducted using the Great Britain (GB) cattle movement database, a government funded system managed by the Department of Environment, Food and Rural Affairs (DEFRA), introduced in 1998. The database records the movement of all cattle between locations in GB.

The contributions of this paper may thus be summarised as follows: (i) a mechanism for identifying exact banding in zero-one data sets, (ii) a mechanism for applying banding to very large data sets using a sampling technique and (iii) the application of the proposed mechanism to a real life data set. The rest of this paper is structured as follows. Section II presents a brief background review of some relevant work. Section III then presents the exact (multiple dots) banding score concept. Section IV presents the proposed END BPM algorithm that utilises the proposed exact banding score mechanism. This is followed in Section V by a worked example illustrating the operation of the END BPM algorithm in the context of a simple 2D data set that features multiple dots. Section VI then presents a description of the proposed END BPM sampling technique. The evaluation of the proposed approaches is presented in Section VII. Finally, some conclusions are given in Section VIII.

II. Related Work

The idea of identifying banded patterns in data was first proposed in [1], [2]. Early work [1], [3] was entirely directed at finding patterns in 2D data, and focussed very much on using heuristics to identify permutations in the data. Two examples are: (i) the Minimum Banded Augmentation (MBA) algorithm [4] and (ii) the Barycentric (BC) algorithm [2].

The MBA algorithm focused on minimizing the distance of non-zero entries from the leading diagonal of a matrix by reordering the original matrix. Two variations of the MBA algorithm were proposed: "Fixed Permutation" (FP) and "Bidirectional Fixed Permutation" (BFP). The MBA-FP algorithm operated by "flipping" zero entries to one entries, while the MBA-BFP algorithm operated in a vice versa manner. Note also that the MBA algorithms (both versions) fixed the column permutations of the data matrix before executing the algorithms [3]. The basic idea was to solve optimally the consecutive one property on the permuted matrix M and ensure that all row intervals would be pairwise overlapping, by going through all the extra rows and making them consecutive. As the fixed column permutation assumption was not a very realistic assumption with respect to many real world situations, heuristic methods were subsequently proposed in [3] to determine a suitable fixed column permutation.

The Barycentric (BC) algorithm [2] was originally proposed in the context of graph drawing, but has subsequently been used in the context of banded patterns. The BC approach produced good results, but proved difficult to scale up to encompass ND data because of the exponential increase in the number of permutations that need to be considered.

The idea proposed in this paper, as already noted, is to use a "banding score" mechanism to iteratively rearrange the elements in each dimension instead of considering large numbers of permutations. In the authors' previous work, relevant work with respect to the banding score idea [5], [9] has already been noted in Section I above.

III. The Exact Banding Score Mechanism

The identification of bandings in ND zero-one data requires the indexes in each dimensions and, in a given data set to be rearranged so as to reveal a best banding (best here is defined in terms of the overall average distance of dots from the leading diagonal). The basic idea advocated in this paper is to iteratively reorder the indexes in each dimension according to an exact banding score until a "best" banding is arrived at. The exact banding score calculation mechanism, assuming the potential for more than one dot per location, is presented in this section.

Prior to considering the exact banding score mechanism a formal definition of the problem domain is required. The data space of interest is considered to comprise a set DIM of n dimensions, $DIM = \{dim_1, dim_2, \ldots, dim_n\}$. Note that the dimensions are not necessarily of equal size. Each dimension dim_i comprises a sequence of k index values $\{e_{i1}, e_{i2}, \ldots, e_{ik}\}$ where i is the dimension identifier. The notation e_{ij} thus indicates an item e in dimension dim_i with index j. Each location contains zero, one or more dots. The precise distribution of the dots depends on the nature of the application domain. Each dot (hypersphere in ND space) will be represented by a set of coordinates dimension (indexes) $\langle c_1, c_2, \ldots, c_n \rangle$ (where n is the number of dimensions) such that $c_1 \in dim_1$, $c_2 \in dim_2$ and so on. The challenge is then to rearrange the indexes in the dimensions so that the dots are arranged along the leading diagonal (or as close to it as possible). Note that, although not demonstrated in this paper, any changes in the dimension ordering does not effect the nature of the resulting banding with respect to any of the algorithms considered in this paper.

The banding score bs_{ij} for an index e_{ij} in DIM_i is calculated by first identifying the sets of locations D_{ij} in the space where "dots" exist that feature index j for dimension dim_i ($D_{ij} = \{d_1, d_2, \ldots\}$). Once the relevant set of dots has been identified the dots in D_{ij} need to be weighted with respect to their proximity to the zero location in the data space (the location within the space where all indexes have the value 0) excluding the current dimension i. The weightings are equivalent to the distance of each dot from the origin. The set of weightings is given by $W_{ij} = \{w_1, w_2, \ldots\}$ such that there is a one to one correspondence between the items in the set W_{ij} and the set $D_{ij} = \{d_1, d_2, \ldots\}$. The calculation of the weightings can be done using a number of mechanisms; the simplest and most natural, and that which was used with respect to this paper, is Euclidean distance. Thus the weighting e_k for a dot at location d_k will be given by:

$$w = \sqrt{\sum_{i=1}^{i=n, i!=j} (c_i)^2} \quad (1)$$

An alternative is to use Manhattan distance which, although not as precise, requires less computation.

To take into consideration the number of dots at each location we also contrive a set $Q_{ij} = \{q_1, q_2, \dots\}$ where q_k holds a "number of dots" value. There is also a one-to-one correspondence between Q_{ij} and D_{ij} (and consequently W_{ij}); thus q_1 corresponds to d_1, q_2 to d_2 ad so on.

The banding score bs_{ij} for index e_{ij} in DIM_i is calculated by summing the distances, each multiplied by the appropriate q_k value, and normalising this by dividing by the maximum possible value for the current configuration. More specifically:

$$bs_{ij} = \frac{\sum_{p=1}^{p=|W_{ij}|} w_p * Q_p}{\sum_{q=1}^{q=|M_{ij}|} m_q * Q'_q} \quad (2)$$

where: (i) W_{ij} is the set of weightings calculated as described above; (ii) Q is the corresponding set of location quantities (see above); (iii) M_{ij} is the set of *maximum weightings* corresponding to the number of dots in D_{ij}, $M_{ij} = \{m_1, m_2, \dots\}$ ($|M_{ij}| = |D_{ij}|$), in descending order; (iv) w_p is the weightings p in W_{ij}, (v) m_q is the maximum weightings q in M_{ij} and (vi) Q'_q is the the set of location quantities Q but in descending order. The calculation of the values for M should be done using the same mechanism as used to calculate the values for W, thus Euclidean distance in the context of this paper. Note that the calculation of maximum weightings will need to be done repeatedly for each index j in each dimension dim_i, thus it might be expedient to do these calculations once and store them in a "maximum weighting table". Thus, given the above, if every location with index j in a dimension i is filled with a single dot the banding score bs_{ij} will be 1.0 (the maximum banding score), if there is only a dot at the origin the banding score will be 0.0 (the minimum banding score).

Once we have the banding scores for for all indexes j in a dimension i we can rearrange the indexes in ascending order of banding score. We can then repeat this operation for the next dimension and so on. Once all the dimensions have been rearranged in this manner we can calculate a global banding score, gbs, for the resulting configuration as follows:

$$gbs = \frac{\sum_{i=1}^{i=|DIM|} \sum_{j=1}^{j=|dim_i|} bs_{ij}}{\sum_{k=1}^{k=|DIM|} |dim_k|} \quad (3)$$

Having rearranged all the indexes in all the dimensions once does not necessarily mean that we have arrived at a best global banding score. We can therefore repeat the entire operation and continue to do so until the gbs value has been maximised (or some maximum number of iterations has been reached).

Note that in the evaluation presented in Section VII, an independent Average Distance (AD) measure is used to determine the "goodness" of a banding; this is calculated as follows:

$$AD = \frac{\sum_{i=1}^{i=|D|} distance\ d_i\ from\ leading\ diagonal}{|D|} \quad (4)$$

IV. The Exact ND Banded Pattern Mining (END BPM) Algorithm

The END BPM algorithm is presented in Algorithm 1. Note that the algorithm includes an input value max, this is used to set a limit on the number of iterations. After each iteration of the entire data space the current global banding score, gbs', is calculated (line 18), and compared with the global banding score attained so far (line 19). If gbs' is greater than gbs we exit with the configuration from the previous iteration (line 20). Otherwise we set gbs to gbs' (line 22), and DIM to DIM' (line 23), and repeat. We continue in this manner until a best gbs value is arrived at or the pre-specified maximum number of iterations is reached. The result is the data space D reconfigured according to the redefined set DIM.

Algorithm 1 The END BPM Algorithm

```
1:  Input
2:  D = Binary valued input data set
3:  DIM = The set of indexes, one set per dimension
4:  max = The maximum number of iterations
5:  Output
6:  D' = The original data set D rearranged so as to display
    as near a banding as possible
7:  gbs = 1 (The global banding score so far)
8:  counter = 0
9:  while counter < max do
10:   for all dim_i ∈ DIM do
11:     for j = 1 to j = |dim_i| do
12:       bs_ij = Banding score for element j in |dim_i|
          calculated using Equation 2
13:     end for
14:     dim'_i = The set dim_i rearranged according to the
        calculated bs (in ascending order)
15:   end for
16:   DIM' = {dim'_1, dim'_2, ..., dim'_n}
17:   D' = Data set D rearranged according to DIM'
18:   gbs' = new gbs for DIM' calculated using Equation 3
19:   if gbs' > gbs then
20:     break
21:   else
22:     gbs = gbs'
23:     DIM = DIM'
24:     D = D'
25:   end if
26:   counter = counter + 1
27: end while
```

V. Worked Example

The operation of the proposed END BPM algorithm is illustrated in this section using a worked example. Given the 2 dimensional 4×4 configuration given in Figure 3; the configuration features $Dim = \{x, y\}$, $dim_x = \{0, 1, 2, 3\}$ and $dim_y = \{0, 1, 2, 3\}$ with multiple dots in some cells. The input D to the END BPM algorithm is thus:

$$D = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \langle 2,3\rangle, \langle 2,3\rangle, \langle 3,0\rangle\}.$$

Considering dimension x first, we calculate the banding scores (taking into account the number of dots per location)

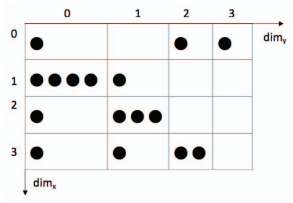


Fig. 3. Input "Dot matrix" for worked example

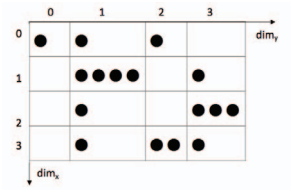


Fig. 4. Dot matrix after rearrangement of dim_x (iteration 1)

using Equation 2. This produces the banding scores 0.60, 0.83, 0.75 and 0.00, calculated as shown in Table I. We thus rearrange the elements in dim_x in ascending order of their banding scores to produce the result shown in Figure 4.

$$D = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \langle 2,3\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,2\rangle, \langle 3,2\rangle, \langle 3,3\rangle\}.$$

# Element	Dist from origin	Max. dist. from origin	bs
0	$(0*1)+(1*4)$ $+(2*1)+(3*1)$ $=9.0$	$(0*1)+(1*1)$ $+(2*1)+(3*4)$ $=15.0$	0.60
1	$(1*1)+(2*3)$ $+(3*1)=10.0$	$(1*1)+(2*1)$ $+(3*3)=12.0$	0.83
2	$(0*1)+(3*2)$ $=6.0$	$(2*1)+(3*2)$ $=8.0$	0.75
3	$(0*1)=0.0$	$((3*1)=3.0$	0.00
		Total	2.18

TABLE I. CALCULATION OF BANDING SCORES FOR DIMENSION x (ITERATION 1)

# Element	Dist from origin	Max. dist. from origin	bs
0	$(0*1)+(1*1)$ $+(2*1)=3.0$	$(1*1)+(2*1)$ $+(3*1)=6.0$	0.50
1	$(1*4)+(3*1)$ $=7.0$	$(2*1)+(3*4)$ $=14.0$	0.50
2	$(1*1)+(3*3)$	$(2*1)+(3*3)$	0.91
2	$=10.0$	$=11.0$	
3	$(1*1)+(2*2)$ $+(3*1)=8.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.89
		Total	2.80

TABLE II. CALCULATION OF BANDING SCORES FOR DIMENSION y (ITERATION 1)

Considering dimension y next, we calculate the banding scores as shown in Table II. This produces the banding scores 0.50, 0.50, 0.91 and 0.89. The elements in y are more or less already in ascending order of bs; we only need to swap the last two elements (the effect is that the index with the greater number of dots is moved to be nearer the centre of the data space). The result is as shown in Figure 5. We now have:

$$D' = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \langle 2,2\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

The gbs for this configuration is then calculated using Equation 3 (the sum of the individual banding scores divided by the total number of indexes in the configuration):

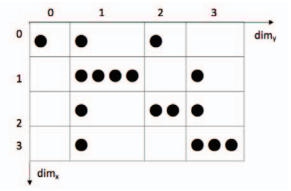


Fig. 5. Dot matrix after rearrangement of dim_y (iteration 1)

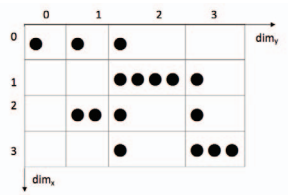


Fig. 6. Dot matrix after rearrangement of dim_x (iteration 2)

$$gbs' = \frac{0.0}{3.0} + \frac{9.0}{15.0} + \frac{6.0}{8.0} + \frac{10.0}{12.0} + \frac{3.0}{6.0} + \frac{7.0}{14.0} + \frac{8.0}{9.0} + \frac{10.0}{11.0} = 0.6122$$

$gbs' < gbs$ (gbs was set to 1 on start up) thus $gbs = gbs'$ and $D = D'$.

The process is then repeated because we have reduced the gbs value and because the maximum number of iterations has not yet been reached. Thus the new banding scores of 0.00, 0.60, 0.50 and 1.00 are produced for dimension x calculated as shown in Table III, and we rearrange the elements in x accordingly; the result is as shown in Figure 6. Similarly, new banding scores of 0.50, 0.79, 0.78 and 1.00 are produced for dimension y calculated as shown in Table IV, as a result the elements in y rearranged accordingly. The result is as shown in Figure 7 (we only needed to swap the second and third indexes). We now have:

$$D' = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 2,0\rangle, \langle 2,1\rangle, \langle 2,1\rangle, \langle 2,1\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

# Element	Dist from origin	Max. dist. from origin	bs
0	$(0*1)=0.0$	$(3*1)=3.0$	0.00
1	$(0*1)+(1*4)$ $+(2*1)+(3*1)$ $=9.0$	$(0*1)+(1*1)$ $+(2*1)+(3*4)$ $=15.0$	0.60
2	$(0*1)+(2*2)$ $=4.0$	$(1*2)+(3*2)$ $=8.0$	0.50
3	$(1*1)+(2*1)$ $(3*3)=12.0$	$(1*1)+(2*1)$ $(3*3)=12.0$	1.00
		Total	2.10

TABLE III. CALCULATION OF BANDING SCORES FOR DIMENSION x (ITERATION 2)

The new gbs' value is calculated as follows:

$$gbs = \frac{0.0}{3.0} + \frac{4.0}{8.0} + \frac{9.0}{15.0} + \frac{12.0}{12.0} + \frac{3.0}{6.0} + \frac{7.0}{9.0} + \frac{11.0}{14.0} + \frac{11.0}{11.0} = 0.6392$$

The $gbs' = 0.6392$ is greater (worse) than the previously calculated value of $gbs = 0.6122$, so the algorithm exits with D from the previous iteration.

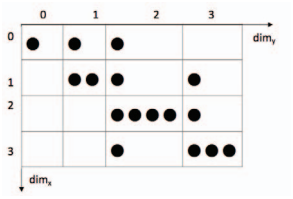


Fig. 7. Dot matrix after rearrangement of dim_y (iteration 2)

$$D = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,0\rangle, \langle 2,2\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 3,3\rangle, \langle 3,3\rangle\}.$$

# Element	Dist from origin	Max. dist. from origin	bs
0	$(0*1)+(1*1)$ $+(2*1)=3.0$	$(1*1)+(2*1)$ $+(3*1)=6.0$	0.50
1	$(2*4)+(3*1)$ $=11.0$	$(2*1)+(3*4)$ $=14.0$	0.79
2	$(1*2)+(2*1)$ $+(3*1)=7.0$	$(1*1)+(2*1)$ $+(3*2)=9.0$	0.78
3	$(2*1)+(3*3)$ $=11.0$	$(2*1)+(3*3)$ $=11.0$	1.00
		Total	3.07

TABLE IV. CALCULATION OF BANDING SCORES FOR DIMENSION y (ITERATION 2)

VI. END BPM WITH SAMPLING

As will be demonstrated in the evaluation section presented later in this paper. The END BPM algorithm works well and produces a better banding than the MBA-FP and MBA-BFP [4] and BC [2] algorithms. However, processing large data sets (data sets that will not fit into primary storage) remains a challenge, especially when the data sets under consideration comprised four or more dimensions. The proposed solution is to adopt a sampling technique where we identify a best banding using a subset S of the dot records in D as shown in algorithm 2. The challenge when using sampling is to select an appropriately representative subset of the original dataset. From the literature we can identify a number of different sampling techniques; however, with respect to the evaluation presented later in this work we have adopted a stratified sampling technique [12], [13] where we divide the data sets into subgroups and select records from each subgroup.

Algorithm 2 END BPM with Sampling Algorithm

1: **Input**
2: $D =$ Binary valued input data set
3: $DIM =$ the set of indexes per dimension
4: $max =$ The maximum number of iterations
5: **Output**
6: $D' =$ The original data set D rearranged so as to display as near a banding as possible and gbs'
7: $S =$ A subset of records from D
8: $S' = end\_bpmAlgorith(S, DIM, max)$ (Algorithm 1)
9: $D' =$ Data set D rearranged according to S'
10: $gbs' =$ Final global banding score calculated using Equation 3

VII. EVALUATION

This section presents an evaluation and discussion of the proposed END BPM algorithm with and without sampling. All the reported experiments were conducted using data extracted from the Great Britain (GB) Cattle Tracking System (CTS). This system is therefore described in Sub-section VII-A. All experiments were run using an iMac running OSX with 16 GB Memory and a 2.7 GHz Intel Core i5 Processor. The algorithms were implemented in JAVA. The first set of reported experiments, presented in Sub-section VII-B, compared the operation of the END BPM [14] algorithm with the AND BPM [5], MBA-FP and MBA-BFP [4] and the BC [2] algorithms from the literature (see Section II). Because MBA-FP, MBA-BFP and BC were designed with respect to 2D data, the comparison was conducted in the 2D context. The second set of reported experiments, presented in Sub-section VII-C compared the operation of the END BPM algorithm with the AND BPM in the context of large ND data. The third set of reported experiments considered the application of the END BPM sampling algorithm in the context of large ND data. The results obtained from these experiments are presented in Sub-section VII-D. To determine the quality of a banding an independent measure was used, as opposed to gbs or the measures used by the MBA-FP, MBA-BFP and BC algorithms, namely the average distance of dots from the leading diagonal; AD measure introduced previously (Equation 4).

A. The GB Cattle Tracking System

The GB CTS was setup in 1998 in response to a cattle disease outbreak. The system is maintained by the British Cattle Movement Service (BCMS), a branch of the UK Department of Environment, Food and Rural Affairs (DEFRA). Central to the system is the cattle movement database which records all movement of cattle in GB. The database can be conveniently divided into blocks of one month, each comprising some 100MB of data. The database features details concerning the "sender" location, "receiver" location and the animal moved. The data can thus be divided according to the geogrphic location of the sender or receiver; for the evaluation presented here we used the counties in which individual senders were located. The database describes individual cattle movements, but cattle are typically moved in batches. Therefore we collapsed records describing cattle movements that occurred on the same day, and with respect to the same sender and receiver locations and the same breed of cattle. To do this we added an extra attribute "number of animal moved". For the evaluation presented here datasets for the years 2003 to 2006 were used.

For the 2D evaluation, presented in Sub-section VII-B below, we only used the CTS data for the year 2003. The two dimensions were *attributes* and *records*. In total 16 data sets were generated by considering four counties (Aberdeenshire, Cornwall, Lancashire and Norfolk) and dividing the year into four quarters. The datasets each comprised six attributes: (i) animal gender, (ii) animal age, (iii) cattle type (dairy or beef), (iv) the "type" of the sender location, (v) the "type" of the receiver location and (vi) the number of cattle moved. These were either nominal or continuously valued attributes and thus had to be translated into a zero-one format. This was done using the LUCS-KDD ARM DN discretisation-normalisation software[1]. Some statistics concerning this data set are presented in Table VI.

For the large scale study presented in Sub-section VII-D, we constructed 3D, 4D and 5D data sets from the CTS

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

database, featuring the four counties considered in the previous experiments. In the case of the 3D data set the three dimensions were: (i) records and (ii) attributes (as for the 2D data sets) and (iii) time in months. For the 4D data sets the four dimensions were: (i) records, (ii) attributes (as for the 2D data sets), (iii) eastings and (iv) northings. For the 5D data set the dimensions were: (i) records, (ii) attributes, (iii) "eastings" (x coordinates of holding areas), (iv) "northings" (y coordinates of holding areas) and (v) time in months. The eastings and northings were discretised into 10 ranges. Some statistics concerning this data set are presented in Tables VII , VIII and IX. To construct the sample set S we considered the individual months across the four identified counties. This gave us $12 \times 4 \times 4 = 192$ data subsets from which we randomly selected 2000 records per subset to give our 3D, 4D and 5D sample sets comprising 24000 records each ($192 \times 24000 = 4,608,000$). Some statistics concerning this sample data sets are presented in Tables X, XI and XII respectively.

It should perhaps also be noted here that there have been a number of previous studies directed at the CTS database. Green and Kao [15] conducted an analysis of the CTS database confirming that as the distance between holding area locations increases the number of movement records decreases. Also Puteri et al. [16] confirmed that by applying trend mining techniques to the CTS data, when envisioned in terms of a social network, trends describing cattle movement across time and geographcal space could be identified. Robinson and Christley [17] also identify a number of trends in the CTS database, demonstrating that the UK cattle population are in constant flux.

TABLE V. NOTATION USED THROUGHOUT THIS PAPER

Notation	Description
D_{ij}	The set of dots associated with dimension i and associted value j
Q_{ij}	number of dots at each location
M_{ij}	The set of *maximum weightings* corresponding to the number of dots in D_{ij}
W_{ij}	The set of weightings $\{w_1, w_2, \dots\}$
w_p	The weightings p in W_{ij}
m_q	The maximum weightings q in M_{ij}

Table V presents the notation used throughout this paper.

TABLE VI. NUMBER RECORDS PER CTS DATA SET USED FOR THE 2D EVALUATION PRESENTED IN SUB-SECTION VII-B

Counties	Q1	Q2	Q3	Q4
Abd	42962	46187	41181	47842
Corn	40501	39626	40226	49890
Lanc	34325	40926	45276	47392
Norf	11526	14311	9460	11680

B. 2D Study and Comparison

For the 2D experiments we recorded the AD measure (Equation 4) and the runtime. In each case we set the max value to 10. The results are presented in Tables XIII and XIV. In Table XIII both the AD independent metric and the gbs metric are reported. Inspection of Table XIII indicates that the END BPM and AND BPM algorithms produce better gbs values (note that the best gbs value is 1.0) and AD results than the other banding algorithms considered. In Table XIV the run time (in seconds) is presented. From this table it can clearly be see that the MBA-BFP, MBA-FP and BC algorithms require considerably more processing time than the proposed END BPM algorithm and the existing approximate AND BPM algorithm. The AND BPM algorithm is faster because, as the name suggests, it makes approximations in the banding calculation that do not manifest themselves unless considering 3D data and above. The main point to note here is that the proposed END BPM and AND BPM algorithm produces better bandings than the MBA-BFP, MBA-FP and BC algorithms.

TABLE VII. NUMBER OF ITEMS (INDEXES) PER DIMENSION FOR THE 16 3D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Time
Aberdeenshire	2003	178172	95	12
	2004	173612	95	12
	2005	157033	95	12
	2006	236206	95	12
Cornwall	2003	170243	98	12
	2004	169053	98	12
	2005	154569	98	12
	2006	167281	98	12
Lancashire	2003	167919	94	12
	2004	217566	94	12
	2005	157142	94	12
	2006	196292	94	12
Norfolk	2003	46977	95	12
	2004	46246	95	12
	2005	35914	95	12
	2006	45150	95	12

TABLE VIII. NUMBER OF ITEMS PER DIMENSION FOR THE 16 4D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.
Aberdeenshire	2003	178172	95	10	10
	2004	173612	95	10	10
	2005	157033	95	10	10
	2006	236206	95	10	10
Cornwall	2003	170243	98	10	10
	2004	169053	98	10	10
	2005	154569	98	10	10
	2006	167281	98	10	10
Lancashire	2003	167919	94	10	10
	2004	217566	94	10	10
	2005	157142	94	10	10
	2006	196292	94	10	10
Norfolk	2003	46977	95	10	10
	2004	46246	95	10	10
	2005	35914	95	10	10
	2006	45150	95	10	10

TABLE IX. NUMBER OF ITEMS PER DIMENSION FOR THE 16 5D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time
Aberdeenshire	2003	178172	95	10	10	12
	2004	173612	95	10	10	12
	2005	157033	95	10	10	12
	2006	236206	95	10	10	12
Cornwall	2003	170243	98	10	10	12
	2004	169053	98	10	10	12
	2005	154569	98	10	10	12
	2006	167281	98	10	10	12
Lancashire	2003	167919	94	10	10	12
	2004	217566	94	10	10	12
	2005	157142	94	10	10	12
	2006	196292	94	10	10	12
Norfolk	2003	46977	95	10	10	12
	2004	46246	95	10	10	12
	2005	35914	95	10	10	12
	2006	45150	95	10	10	12

C. Comparison Between END BPM and AND BPM Algorithms

This subsection presents evaluation of the proposed END BPM in comparison with the AND BPM algorithms, in terms

TABLE XIII. 2D BANDING EVALUATION RESULTS IN TERMS OF AD AND gbs, FOR THE FIVE BANDING MECHANISMS CONSIDERED (BEST RESULTS IN BOLD FONT)

Data Set	END BPM		AND BPM		MBA-FP		MBA-BFP		BC	
	AD	gbs	AD	gbs	AD	gbs	AD	gbs	AD	gbs
42962	**0.9995**	**0.7762**	**0.9995**	**0.7717**	0.9996	0.7076	1.0000	0.7266	0.9998	0.6756
46187	**0.9996**	**0.7463**	**0.9996**	**0.7267**	0.9999	0.7085	1.0000	0.7068	0.9998	0.6575
41181	**0.9995**	**0.7675**	**0.9995**	**0.7447**	0.9996	0.6867	1.0000	0.7267	0.9997	0.6508
47842	**0.9995**	**0.7696**	**0.9995**	0.7395	0.9999	0.7285	1.0000	0.6307	0.9997	0.6652
40501	**0.9994**	**0.7711**	**0.9994**	**0.7666**	0.9995	0.6781	0.9998	0.7438	0.9996	0.6972
39626	**0.9993**	**0.7595**	**0.9993**	**0.7317**	0.9993	0.7048	0.9999	0.7307	0.9996	0.7107
40226	**0.9994**	**0.7713**	**0.9994**	**0.7470**	0.9995	0.7065	0.9999	0.7355	0.9997	0.7045
49890	**0.9995**	**0.7484**	**0.9995**	**0.7397**	0.9996	0.6940	0.9999	0.6993	0.9997	0.6857
34325	**0.9993**	**0.7684**	**0.9993**	0.7376	0.9997	0.7630	0.9999	0.7556	0.9995	0.7059
40926	**0.9994**	**0.7730**	**0.9994**	**0.7595**	0.9995	0.7575	0.9999	0.7350	0.9997	0.7167
45276	**0.9995**	**0.7823**	**0.9995**	0.7114	0.9996	0.7536	0.9999	0.6993	0.9996	0.6857
47392	**0.9995**	**0.7725**	**0.9995**	0.7635	0.9997	0.7696	1.0000	0.7661	0.9999	0.7101
11526	**0.9984**	**0.7504**	**0.9984**	**0.7370**	0.9999	0.6974	0.9999	0.7072	0.9989	0.6996
14311	**0.9985**	**0.7807**	**0.9985**	**0.7449**	0.9986	0.7035	0.9995	0.7338	0.9987	0.6942
9460	**0.9976**	**0.7784**	**0.9976**	**0.7325**	0.9977	0.7188	0.9986	0.7314	0.9983	0.7165
11680	**0.9982**	**0.7379**	**0.9982**	**0.7395**	0.9999	0.7351	0.9999	0.7091	0.9985	0.6921

TABLE X. NUMBER OF SAMPLED DATA ITEMS PER DIMENSION FOR THE 16 3D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Time
Aberdeenshire	2003	24000	95	12
	2004	24000	95	12
	2005	24000	95	12
	2006	24000	95	12
Cornwall	2003	24000	98	12
	2004	24000	98	12
	2005	24000	98	12
	2006	24000	98	12
Lancashire	2003	24000	94	12
	2004	24000	94	12
	2005	24000	94	12
	2006	24000	94	12
Norfolk	2003	24000	95	12
	2004	24000	95	12
	2005	24000	95	12
	2006	24000	95	12

TABLE XI. NUMBER OF SAMPLED DATA ITEMS PER DIMENSION FOR THE 16 4D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.
Aberdeenshire	2003	24000	95	10	10
	2004	24000	95	10	10
	2005	24000	95	10	10
	2006	24000	95	10	10
Cornwall	2003	24000	98	10	10
	2004	24000	98	10	10
	2005	24000	98	10	10
	2006	24000	98	10	10
Lancashire	2003	24000	94	10	10
	2004	24000	94	10	10
	2005	24000	94	10	10
	2006	24000	94	10	10
Norfolk	2003	24000	95	10	10
	2004	24000	95	10	10
	2005	24000	95	10	10
	2006	24000	95	10	10

TABLE XII. NUMBER OF SAMPLED DATA ITEMS PER DIMENSION FOR THE 16 5D CTS DATA SETS

Counties	Years	# Recs.	# Atts.	# Eings.	# Nings.	# Time
Aberdeenshire	2003	24000	95	10	10	12
	2004	24000	95	10	10	12
	2005	24000	95	10	10	12
	2006	24000	95	10	10	12
Cornwall	2003	24000	98	10	10	12
	2004	24000	98	10	10	12
	2005	24000	98	10	10	12
	2006	24000	98	10	10	12
Lancashire	2003	24000	94	10	10	12
	2004	24000	94	10	10	12
	2005	24000	94	10	10	12
	2006	24000	94	10	10	12
Norfolk	2003	24000	95	10	10	12
	2004	24000	95	10	10	12
	2005	24000	95	10	10	12
	2006	24000	95	10	10	12

TABLE XIV. RUNTIME (IN SECONDS) FOR THE FIVE BANDING MECHANISMS CONSIDERED (BEST TIMES IN BOLD FONT)

Data Set	END BPM	AND BPM	MBA-FP	MBA-BFP	BC
42962	**30.40**	**28.74**	64.59	60.62	51.36
46187	**52.32**	**51.17**	76.77	65.40	54.60
41181	**26.48**	**25.89**	62.42	59.41	43.74
47842	**35.03**	**34.98**	79.31	75.46	61.94
40501	**24.95**	**23.00**	56.69	60.31	41.36
39626	**13.27**	**12.55**	64.01	56.88	39.80
40226	**28.61**	**26.82**	62.42	60.84	40.10
49890	**47.50**	**47.03**	89.17	82.54	61.15
34325	**19.61**	**18.62**	49.31	50.34	21.76
40926	**27.99**	**26.17**	58.08	60.08	34.61
45276	**41.39**	**40.53**	75.61	69.78	46.57
47392	**43.83**	**42.86**	77.91	78.97	50.69
11526	**02.77**	**02.54**	09.92	10.47	03.85
14311	**03.26**	**02.27**	13.84	12.68	05.34
9460	**02.03**	**01.98**	07.89	07.85	02.86
11680	**02.66**	**02.60**	10.42	10.44	03.53

of gbs values and run-times, using the 16 datasets from the GB cattle movement database in the context of: (i) 3D data banded in terms of 2D, (ii) 4D data banded in terms of 3D and (iii) 5D data banded in terms of 4D. Table XV and Table XVI report the gbs values (here the best gbs value is 0.0) and the run time (in seconds). From Table XV, the result confirms that the END BPM algorithm is more accurate and more effective than the AND BPM algorithm, for 3D data and above. Although in terms of the run-time results presented in Table XVI, the result shows that the END BPM algorithm is slower than the AND BPM algorithm. Note that as stated in Sub-section VII-B, the AND BPM algorithm produces an approximate banding calculation that manifests only when considering 3D datasets and above. Table XV confirms that the gbs values for the 3D dataset, banded in terms of 2D, are the same for both the END BPM and AND BPM algorithms. However in the case of the 4D and 5D datasets, the result shows that the AND BPM algorithm produced the worst gbs values.

D. ND Study and Comparison

This subsection presents an evaluation of the END BPM algorithm in terms of ND data, and in terms of sampling. With respect to the ND experiments using sampling it should be

TABLE XV. ND BANDING RESULTS, IN TERMS OF gbs WHEN USING THE END BPM AND AND BPM ALGORITHMS

Counties /	END BPM			AND BPM		
Year id	$3D$	$4D$	$5D$	3D	4D	5D
Aberdeenshire	gbs	gbs	gbs	gbs	gbs	gbs
2003	**0.3770**	**0.2906**	**0.2340**	0.3770	0.4006	0.4226
2004	**0.3686**	**0.2409**	**0.2020**	0.3686	0.3526	0.3814
2005	**0.3869**	**0.3080**	**0.2482**	0.3869	0.4155	0.4470
2006	**0.3670**	**0.2380**	**0.1988**	0.3670	0.3449	0.3780
Cornwall						
2003	**0.4039**	**0.2903**	**0.2401**	0.4039	0.4180	0.4426
2004	**0.3944**	**0.2594**	**0.2160**	0.3944	0.3743	0.4050
2005	**0.3696**	**0.2708**	**0.2240**	0.3696	0.3943	0.4175
2006	**0.3886**	**0.3065**	**0.2510**	0.3886	0.4336	0.4554
Lancashire						
2003	**0.4105**	**0.2924**	**0.2428**	0.4105	0.4199	0.4429
2004	**0.3593**	**0.2755**	**0.2351**	0.3593	0.3959	0.4179
2005	**0.3974**	**0.2915**	**0.2441**	0.3974	0.4142	0.4437
2006	**0.3970**	**0.2824**	**0.2366**	0.3970	0.4071	0.4369
Norfolk						
2003	**0.4265**	**0.3079**	**0.2663**	0.4265	0.4575	0.4696
2004	**0.3607**	**0.2810**	0.2349	0.3607	0.4005	0.4266
2005	**0.4058**	**0.3179**	**0.2547**	0.4058	0.4422	0.4540
2006	**0.3802**	**0.2529**	**0.2125**	0.3802	0.3634	0.3948

TABLE XVI. RUNTIME (IN SECONDS) WHEN USING THE END BPM AND AND BPM ALGORITHMS

Counties /	END BPM			AND BPM		
Year id	$3D$	$4D$	$5D$	3D	4D	5D
Aberdeenshire	RT	RT	RT	RT	RT	RT
2003	01.58	02.68	24.69	**01.05**	**01.47**	**09.97**
2004	01.92	02.90	26.93	**01.35**	**01.39**	**12.41**
2005	01.85	02.95	15.66	**01.57**	**01.61**	**09.19**
2006	01.48	01.79	18.41	**01.30**	**01.42**	**07.41**
Cornwall						
2003	01.51	01.71	24.09	**01.02**	**01.38**	**09.61**
2004	02.55	03.41	35.62	**01.92**	**01.74**	**19.61**
2005	01.53	02.90	27.43	**01.34**	**02.19**	**11.92**
2006	02.27	06.01	41.89	**01.06**	**02.40**	**25.41**
Lancashire						
2003	01.59	02.97	55.19	**01.26**	**02.07**	**30.02**
2004	01.48	02.91	39.29	**01.06**	**02.15**	**19.09**
2005	01.57	02.62	29.11	**01.30**	**01.87**	**17.84**
2006	01.62	02.43	28.31	**01.20**	**02.37**	**17.49**
Norfolk						
2003	01.17	05.09	20.84	**01.01**	**02.24**	**10.72**
2004	02.99	05.94	80.20	**01.00**	**03.41**	**27.99**
2005	01.15	02.16	16.41	**00.95**	**01.74**	**11.66**
2006	01.52	01.83	17.92	**01.00**	**01.29**	**10.99**

noted that it does not make sense to reorder the records dimension as we are segmenting the data according to this dimension. Thus banding was applied to the remaining dimensions in the sample, hence the 3D data set was banded in terms of 2D, the 4D data set in terms of 3D, and so on. Consequently, it is entirely possible to have more than one dot per location hence the multiple dot banding score mechanism as presented in Section III (note that the banding mechanism will work equally well where we have data sets with a maximum of one dot per location).

The results obtained from the ND experiments are presented in Tables XVII and XVIII. In Table XVII, the gbs metric is reported and in Table XVIII, the run time (in seconds) is presented. In this case, to determine the effectiveness of END BPM, we use the gbs values produced. The table presents gbs values with repect to: (i) the sample data, (ii) the original data set with banding from the sample and (iii) the original data set in its raw form without banding. Table XVII shows the gbs values obtained (note that the best gbs value is 0.0). The columns represent, in order, the four counties (Aberdeenshire, Cornwall, Lancashire and Norfolk), the year and the size of the dataset considered (number of records). From Table XVII, it can thus be seen that by imposing the banding identified in the sample on the entire data set the gbs for the entire data set can be improved. In most cases the sample bandings is better than the eventual global banding (this is to be expected), however in one case Aberdeenshire 2006, we got it exactly right.

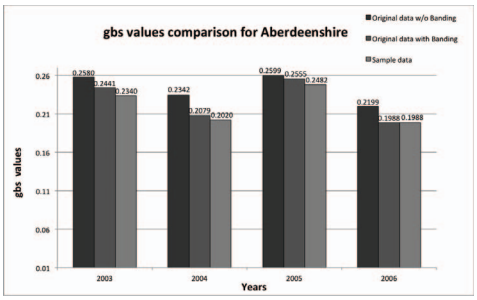
To enhance appreciation of the results obtained, the results from Table XVII is presented in bar graph form in Figure 8. From the graph, it can again be seen that there is an improvement in banding on the original data after applying the banding from the sample data to the original data. Figures 9 and 10 presents the results from Tables XVIII and XVI respectively in bar graph form.

TABLE XVII. EFFECTIVENESS RESULTS, IN TERMS OF gbs, WHEN USING THE END BPM ALGORITHM

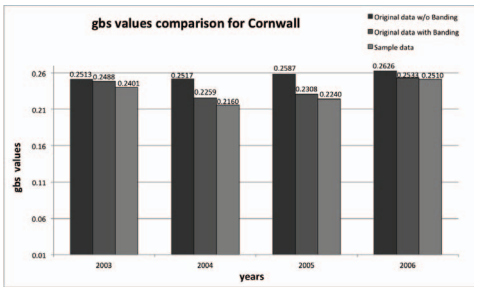
Counties	Year	Dataset	END BPM		
	id	Num. Recs.	$3D$	$4D$	$5D$
Aberdeenshire			gbs	gbs	gbs
Sample data	2003	24000	**0.3770**	**0.2906**	**0.2340**
Original data with Banding		178173	0.3811	0.3033	0.2441
Original data w/o Banding		178173	0.3937	0.3176	0.2580
Sample data	2004	24000	**0.3686**	**0.2409**	**0.2020**
Original data with Banding		173612	0.3710	0.2486	0.2079
Original data w/o Banding		173612	0.3947	0.2924	0.2342
Sample data	2005	24000	**0.3869**	**0.3080**	**0.2482**
Original data with Banding		157033	0.3977	0.3176	0.2555
Original data w/o Banding		157033	0.3987	0.3205	0.2599
Sample data	2006	24000	**0.3670**	**0.2383**	**0.1988**
Original data with Banding		236206	0.3709	**0.2383**	0.1999
Original data w/o Banding		236206	0.3733	0.2716	0.2199
Cornwall					
Sample data	2003	24000	**0.4039**	**0.2903**	**0.2401**
Original data with Banding		170243	0.4048	0.2983	0.2488
Original data w/o Banding		170243	0.4370	0.3052	0.2513
Sample data	2004	24000	**0.3944**	**0.2594**	**0.2160**
Original data with Banding		169053	0.4023	0.2723	0.2259
Original data w/o Banding		169053	0.4043	0.3113	0.2517
Sample data	2005	24000	**0.3696**	**0.2708**	**0.2240**
Original data with Banding		154569	0.3786	0.2781	0.2308
Original data w/o Banding		154569	0.4142	0.3168	0.2587
Sample data	2006	24000	**0.3886**	**0.3065**	**0.2510**
Original data with Banding		167281	0.3901	0.3077	0.2533
Original data w/o Banding		167281	0.4060	0.3134	0.2626
Lancashire					
Sample data	2003	24000	**0.4105**	**0.2924**	**0.2428**
Original data with Banding		167919	0.4136	0.3046	0.2531
Original data w/o Banding		167919	0.4206	0.3136	0.2547
Sample data	2004	24000	**0.3593**	**0.2755**	**0.2351**
Original data with Banding		217566	0.3769	0.2874	0.2450
Original data w/o Banding		217566	0.3908	0.3148	0.2576
Sample data	2005	24000	**0.3974**	**0.2915**	**0.2441**
Original data with Banding		157142	0.4007	0.2938	0.2446
Original data w/o Banding		157142	0.4059	0.3000	0.2459
Sample data	2006	24000	**0.3970**	**0.2824**	**0.2366**
Original data with Banding		196292	0.3999	0.2830	0.2405
Original data w/o Banding		196292	0.4011	0.3022	0.2459
Norfolk					
Sample data	2003	24000	**0.4255**	**0.3079**	**0.2663**
Original data with Banding		46977	0.4270	0.3103	0.2695
Original data w/o Banding		46977	0.4370	0.3319	0.2717
Sample data	2004	24000	**0.3607**	**0.2810**	**0.2349**
Original data with Banding		46246	0.3653	0.2917	0.2417
Original data w/o Banding		46246	0.3698	0.2993	0.2444
Sample data	2005	24000	**0.4010**	**0.3179**	**0.2547**
Original data with Banding		35914	0.4075	0.3187	0.2557
Original data w/o Banding		35914	0.4148	0.3202	0.2564
Sample data	2006	24000	**0.3802**	**0.2529**	**0.2125**
Original data with Banding		45150	0.3817	0.2554	0.2132
Original data w/o Banding		45150	0.3986	0.2568	0.2142

VIII. CONCLUSIONS

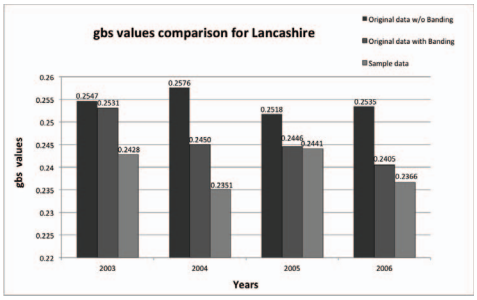
This paper has described a banding mechanism for large scale zero-one data analysis. The aim is to identify bandings in



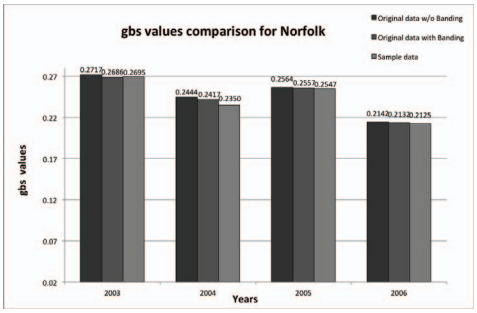
(a)



(b)



(c)

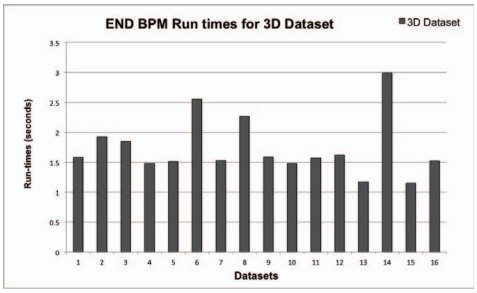


(d)

Fig. 8. GBS values comparison using the Great Britain (GB) cattle movement data for four counties: (a) Aberdeensire, (b) Cornwall, (c) Lancashire and (d) Norfolk.

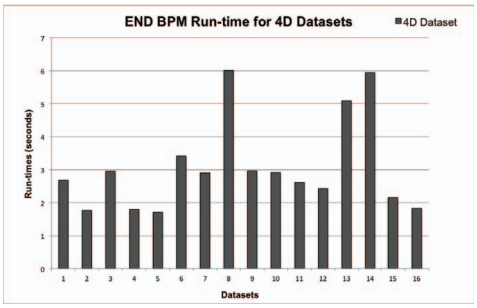
TABLE XVIII. EFFECIENCY RESULTS, IN TERMS OF RUNTIME (SECOND)S, USING END BPM ALGORITHM CONSIDERED (BEST RESULTS IN BOLD FONT)

Counties	Year id	Datasets Num. Recs.	END BPM run-time (seconds) $3D$ RT	$4D$ RT	$5D$ RT
Aberdeenshire					
Sample data	2003	24000	**01.58**	**02.68**	**24.69**
Original data with Banding		178173	14.18	16.08	47.85
Original data w/o Banding		178173	14.02	19.36	45.63
Sample data	2004	24000	**01.92**	**01.76**	**26.93**
Original data with Banding		173612	13.13	15.84	45.80
Original data w/o Banding		173612	13.16	16.20	48.54
Sample data	2005	24000	**01.85**	**02.95**	**15.66**
Original data with Banding		157033	12.98	14.24	47.88
Original data w/o Banding		157033	13.61	15.97	44.28
Sample data	2006	24000	**01.48**	**01.79**	**18.41**
Original data with Banding		236206	14.47	15.87	56.01
Original data w/o Banding		236206	17.22	21.01	47.55
Cornwall					
Sample data	2003	24000	**01.51**	**01.71**	**24.09**
Original data with Banding		170243	14.86	50.87	51.92
Original data w/o Banding		170243	17.73	16.94	46.87
Sample data	2004	24000	**02.55**	**03.41**	**35.62**
Original data with Banding		169053	18.28	22.44	77.18
Original data w/o Banding		169053	15.40	28.43	75.38
Sample data	2005	24000	**01.53**	**02.90**	**27.43**
Original data with Banding		154569	14.32	14.61	43.53
Original data w/o Banding		154569	14.82	21.13	42.04
Sample data	2006	24000	**02.27**	**06.01**	**41.89**
Original data with Banding		167281	14.50	17.6	94.80
Original data w/o Banding		167281	17.63	19.43	92.51
Lancanshire					
Sample data	2003	24000	**01.59**	**02.97**	**55.19**
Original data with Banding		167919	13.94	16.59	104.30
Original data w/o Banding		167919	17.74	18.02	91.92
Sample data	2004	24000	**01.48**	**02.91**	**39.29**
Original data with Banding		217566	15.25	19.99	85.90
Original data w/o Banding		217566	17.97	23.65	82.91
Sample data	2005	24000	**01.57**	**02.62**	**29.11**
Original data with Banding		157142	16.08	16.65	69.82
Original data w/o Banding		157142	15.75	26.76	70.47
Sample data	2006	24000	**01.62**	**02.43**	**28.31**
Original data with Banding		196292	14.72	18.83	82.79
Original data w/o Banding		196292	25.57	22.43	78.59
Norfolk					
Sample data	2003	24000	**01.17**	**05.09**	**20.84**
Original data with Banding		469773	11.73	12.09	20.90
Original data w/o Banding		46977	14.42	14.07	22.64
Sample data	2004	24000	**02.99**	**05.94**	**80.20**
Original data with Banding		46246	16.42	27.05	85.48
Original data w/o Banding		46246	15.71	16.27	83.48
Sample data	2005	24000	**01.15**	**02.16**	**16.41**
Original data with Banding		35914	10.85	11.23	18.69
Original data w/o Banding		35914	16.96	14.55	17.41
Sample data	2006	24000	**01.52**	**01.83**	**17.92**
Original data with Banding		45150	11.10	12.30	21.77
Original data w/o Banding		45150	15.09	15.61	19.98

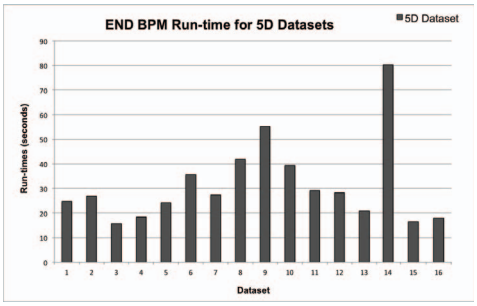


(a)

large data sets using an exact banded pattern mining algorithm. The proposed mechanism has been evaluated using benchmark and sample data sets derived from the Great Britain (GB) cattle movement database. We have shown that the proposed END BPM mechanism is able to identify accurate bandings in large data sets within reasonable computation time. The results presented confirms that the END BPM algorithms works well in ND (in terms of effectiveness). For future work the authors intend to extend their research to address situations where we seek to sequentially establish bandings in large datasets using sequences of data segments. Whatever the case, the authors

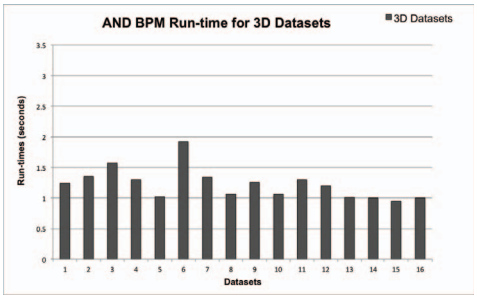


(b)

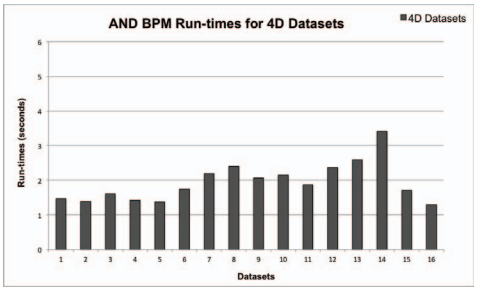


(c)

Fig. 9. END BPM Runtime (in seconds) using the Great Britain (GB) cattle movement sampling dataset in the context of: (a) 3D, (b) 4D and (c) 5D.

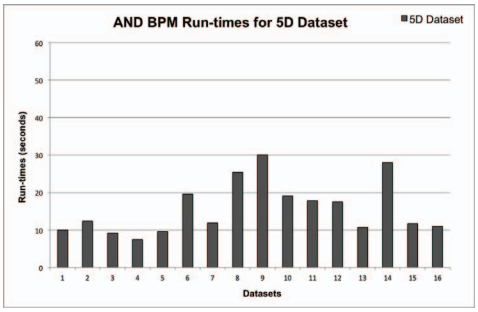


(a)



(b)

have been greatly encouraged by the results produced so far, as presented in this paper.



(c)

Fig. 10. AND BPM Runtime (in seconds) using the Great Britain (GB) cattle movement dataset in the context of: (a) 3D, (b) 4D and (c) 5D.

REFERENCES

[1] H. Mannila and E. Terzi, "Nested and segmented nested." in *Proceedings of the 13h ACM SIGKDD international conference on knowledge discovery and data mining, New York, NY, USA, 2007*, 2007, pp. 480–489.

[2] E. Makinen and H. Siirtola, "The barycenter heuristic and the reorderable matrix." *Informatica*, vol. 29, pp. 357–363, 2005.

[3] G. C. Gemma, E. Junttila, and H. Mannila., "Banded structures in binary matrices," *Knowledge Discovery and Information System*, vol. 28, pp. 197–226, 2011.

[4] E. Junttila, "Pattern in permuted binary matrices," Ph.D. dissertation, 2011.

[5] F. B. Abdullahi, F. Coenen, and R. Martin, "A scalable algorithm for banded pattern mining in multi-dimensional zero-one data," in *In Proc. Data Warehousing and Knowledge Discovery (DaWaK'14). Springer, LNAI*, 2014, pp. 391–404.

[6] K. Y. Cheng, "Minimising the bandwidth of sparse symmetric matrices," in *Computing*, vol. 11, pp. 103–110.

[7] A. E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices." in *Proceedings of the 24th National Conference of ACM*, 1969, pp. 157–172.

[8] C. H. Papadimitrious, "The np-completeness of the bandwidth minimisation problem," in *Computing*, vol. 16, 1976, pp. 263–270.

[9] F. B. Abdullahi, F. Coenen, and R. Martin, "A novel approach for identifying banded patterns in zero-one data using column and row banding scores," in *In Proc. Machine Learning and Data Mining in Pattern Recognition (MLDM). Springer, LNAI*, 2014, pp. 58–72.

[10] J. Atkins, E. Boman, and B. Hendrickson., "Spectral algorithm for seriation and the consecutive ones problem," *SIAM J. Comput.*, vol. 28, pp. 297–310, 1999.

[11] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. Mooney, "Model-based overlapping clustering." in *Proceedings of Knowledge Discovery and DataMining*, 2005, pp. 532–537.

[12] P. S. Levy and S. Lemeshow, "Sampling of populations: Methods and applications," *New York: Wiley and Sons*, 2008.

[13] M. A. Burnam and P. Koegel, "Methodology for obtaining a representative sample of homeless persons: The los angeles skid row study." *Evaluation Review*, vol. 12, pp. 117–52, 1988.

[14] F. B. Abdullahi, F. Coenen, and R. Martin, "Finding banded patterns in data: The banded pattern mining algorithm," in *In Proc. Big Data Analytics and Knowledge Discovery (DaWaK'15). Springer, LNAI*, 2015, pp. 95–107.

[15] D. Green and R. Kao, "Data quality of the cattle tracing system in great britain." *Veterinary Record*, vol. 161, pp. 439–443, 2007.

[16] N. N. Puteri, R. Christley, . ", and C. Setzkorn, "Trend mining in social networks: A study using a large cattle movement database," *Advances in Data mining, Applications and Theoretical Aspects LNCS*, vol. 6171, pp. 464–475, 2010.

[17] S. Robinson and R. Christley, "Identifying temporal variation in reported birth, death and movements of cattle in britain." in *BMC Veterinary Research*, 2006, pp. 2–11.