Grid-Based Spatial Data Compression Scheme for Clustering in Artificial Intelligence

Jongwan Kim
Smith College of Liberal Arts, Sahmyook University,
815 Hwarang-ro, Nowon-gu, Seoul, 01795, Korea
kimj@syu.ac.kr
http://www.syu.ac.kr

Abstract - We propose a grid-based spatial data compression scheme for fast spatial data processing of clustering in machine learning. Existing studies focused on optimizing clustering algorithms centering on spatial data processing; however, the storage utilization and service transfer time were not sufficiently considered. In this study, three spatial data are compressed in an 8-bit string to process more spatial data, thus enabling a faster machine learning time for not only location-based services but also grid-based clustering. In the experiment, we compared the storage rates of 16-byte spatial coordinates and multi-bit compressed coordinates. The experiment result demonstrated that the storage space utilization of compressed spatial data in 4KB blocks was improved by approximately 24 times.

***Keywords*:** Clustering; Location-based service; Spatial data compression; Grid; Bit map.

1. Introduction

A location-based service (LBS) provides information on the location of an infectious disease, such as avian influenza and COVID-19, and the location of emergency disasters along with entertainment, based on the location provided by GPS [Petch (1999)], [Elghazal *et al.* (2017)], [Gartner and Huang (2016)], [Ekong *et al.* (2017)]. The aforementioned services provide geographic location information using maps, and spatial clustering algorithms are used in artificial intelligence to classify infectious diseases, as well as animals and plants, based on spatial characteristics [Zhou *et al.* (2016)]. Spatial clustering is created by grouping spatial objects and learning the characteristics of groups using machine learning. In particular, spatial clustering identifies groups of spatial objects in a particular area, finds related objects, and sends data about the groups to users.

Spatial data analyzed by clustering are stored on the server, hence the size of spatial data affects the quality of service when learning spatial data or sending results. It also affects the storage size of the server.

Clustering of spatial data improves the quality of services by quickly analyzing related objects distributed in space in terms of services provided to users. Existing location-based services and spatial data clustering algorithms have been studied to improve processing speed by reducing the time complexity or amount of codes through algorithm optimization such as spatial data identification technique [Zhang et al. (2018)] or clustering technique [Zhou et al. (2016)].

The spatial data used for clustering are represented as MBR in R-tree [Guttman (1984)], and one object is represented by two points (x1, y1)-(x2, y2) of a diagonal edge, as shown in Fig. 1. The four coordinates of the two points are each 4 bytes in size, which can be a burden in terms of the speed when clustering or transmitting results to the client.

In this study, we propose a spatial data compression scheme, multi-bit compression, using multiple bits to reduce processing time of the scheme in clustering spatial data. The idea is that because spatial data is managed by MBR in spatial indices, and the planes that come in contact with the x-axis and y-axis in a two-dimensional space are then compressed by storing them as bit strings. The MBR information stored increases with the length of the bit string for each axis. As a result, a large number of spatial data is stored in one-bit string, thereby saving server storage space and improving performance in data processing in machine learning.

The contributions of this study are as follows.

- Spatial data compression provides additional performance improvements in addition to algorithm improvements in spatial data clustering using machine learning.
- When transmitting the clustering result to clients, it is possible to provide a quick service by increasing the amount of spatial data included in a single unit.
- Spatial data compression reflects the same characteristics of grid-based clustering.
- A multi-bit compression scheme is proposed in which the compression ratio of spatial data is improved according to the length of the bits constituting the axis in a two-dimensional space.

- The space on the server is saved by compressing spatial data.

The paper is organized as follows. Section 2 examines the spatial data index, R-tree, and Section 3 describes the spatial data compressor, the multi-bit compressor. In Section 4, the experiment is discussed, and the conclusion of this paper is explained in Section 5.

2. Preliminaries

2.1. *Spatial Index*

Geographical spatial objects exist in various forms and have two-dimensional data of latitude and longitude. R-tree is a spatial index proposed in [Guttman (1984)] to manage spatial objects as indices. The spatial index expresses various shapes of spatial data as a minimum bounding rectangle (MBR), and the inclusion relationship between MBRs is represented by a tree structure of *R0 (r1, r2, r3, r4)* as shown in Fig. 1. In other words, the spatial data are organized hierarchically as MBR under *r1*.

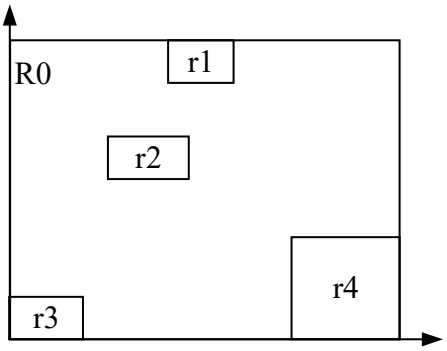

Fig. 1. MBRs in a 2D Space.

2.2. *Grid-Based Spatial Clustering*

Because spatial data are distributed in a certain area, they are called grid-based spatial partitioning, thereby dividing the area into grids and determining the distribution into grid cells occupied by the data [Wu and Wilamowski (2017)].

In machine learning, clustering is a technique of classifying similar data by grouping and classifying them based on their location or attributes in a space. In previous studies, as shown in Fig. 2, grid-based clustering has been used for disease distribution, the location of preferred tourist attractions, and product distribution.

The grid divides the area into cell units, making it easy to judge the area. Therefore, many studies suggest useful research results through grid-based clustering [Jannu and Jana (2016)], [Zhang et al. (2018)]. However, because the location information of the spatial data uses the same original coordinates, the processing speed will be further improved if the spatial data are compressed. In particular, if you compress multiple MBRs at once with multiple compression rather than compressing spatial data individually, you can provide faster results in data processing and services.

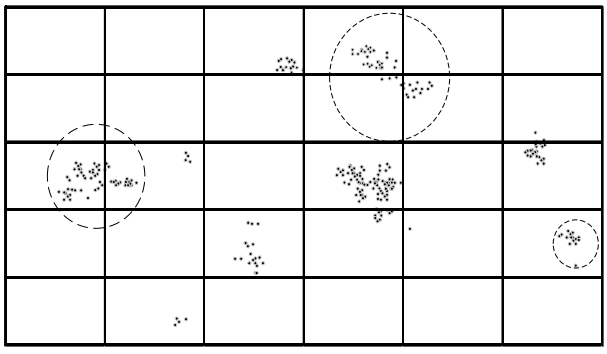

Fig. 2. Spatial data clustering using a grid.

3. Multi-Bit Compression of Spatial Data

Spatial data represented by MBR are shaped like cells of grid, thus 4-byte integer or float type coordinates are converted to the bit level if the position of the data is saved as cell lines on the x- and y-axes. Compressed spatial data can reduce the location computational time when clustering the data in machine learning, and it can transmit large amounts of spatial data at once when sending results to clients.

Fig. 3. shows the clustering of data by machine learning in a two-dimensional space. The area clustered by initial spatial data and machine learning has the location information. If the search area is divided into grids and the top, bottom, left, and right sides of the MBR are replaced with grid lines, the spatial information is reduced to bits. For example, cluster *A*, which occupies six cells, is represented by a rectangle; when the grid line that intersects the x-axis is represented as 1 and the place where it does not intersect the x-axis is represented as 0, it results in 10100000. If clusters *D* and *F* are expressed together with *A*, it becomes 10101011. Therefore, the x and y axes of the clusters *A*, *D*, and *F* are converted into (10101011, 01110100) and represent an MBR which includes three clusters with a total of 16 bits. In the multi-compression technique, even in the case of single spatial data as shown in Fig. 3, the MBR for *Z* can be extended to the surrounding cells and represented as bits. The cells containing *Z* can be divided further to reduce the range of extension.

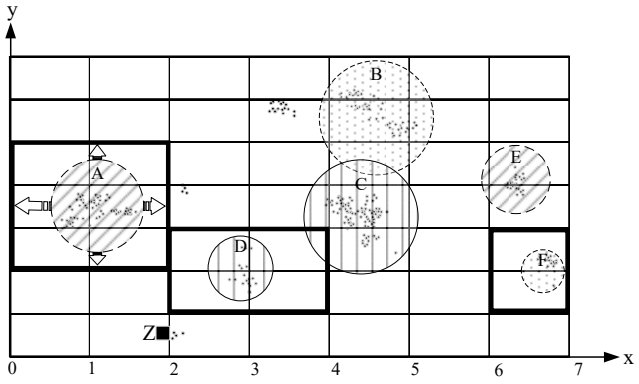

Fig. 3. Grid-Based Clustering and MBRs.

The number of grids that divide the search area depends on the number of bits, and the definition of the length of the bits is as follows.

Definition 1. *The length of a bit string.*

Let the search area be *r*, which represent x- and y-axis. If the extrusion size of the spatial data is called size_bits, the number of grids *g* is as follows. The unit of the grid cell with respect to the search area on one axis of the two-dimensional is 0 ~ (size_bits - 1).

$$g = (r.x \text{ or } r.y) / size\_bits \quad . \tag{1}$$

In grid-based clustering, the characteristics of representing the location of spatial data sets as grid cells are summarized as the following heuristics.

Heuristic 1. *Coordinate Compression by Bit Conversion.*

Dividing the spatial coordinates by the number of bits gives the remainder a value between 0 and (number of bits - 1), which is a grid line representing the area surrounding the MBR or the space containing clustering. Therefore, the position of the spatial data may be represented by a smaller value through the position of the line meeting the grid cell to indicate the area of the spatial data.

Multi-bit compression of spatial coordinates saves the storage space by reducing the storage unit of one spatial data and speeds up processing time of the spatial data. In particular, owing to compression, the clustering results of influenza or specific areas analyzed in machine learning are transmitted quickly over the network.

Fig. 4 shows the restoration of the MBR from multiple compressed bit streams. In the compressed information, the numbers on the x- and y-axes represent 1 if there is a line that intersects each axis, and 0 if not. The axis coordinates of clusters *A*, *D*, and *F* are (10101011, 01110100), and when the line is drawn around 1 where each axis meets a line, the MBR of each cluster can be restored. The bits of cluster *A* are 10100000 in the x-axis and 00100100 in the y-axis. Likewise, if a line is drawn based on the bits of each axis, the areas for clusters *A*, *D*, and *F* are detected.

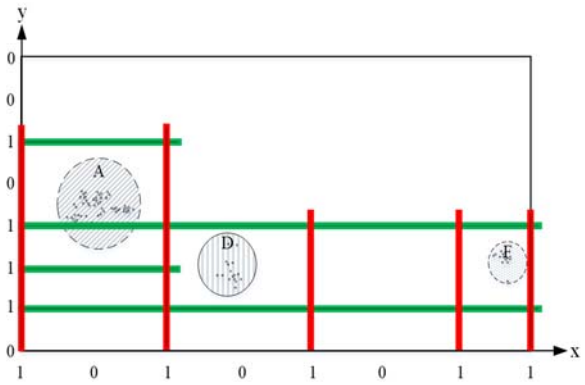

Fig. 4. MBR restore using multi-bit compression.

4. Simulation

The multi-compressor method using the bitmap proposed in this study is to perform a clustering process by compressing the coordinates of spatial data and to increase the utilization of storage space. In the experiment, we generated virtual spatial data and measured the storage space efficiency for MBR representing the area of clustering. The simulation environment is shown in Table 1.

Table 1. Simulation Environment

Category	Contents
Data Set	Virtual Spatial Data 100K
Saving Blocks	512, 1024, 2048, 4096
Data Distribution	Uniform, Skewed
Development Language and System	Python 3.2, Fedora 30, 2 CPU XEON, RAM 64GB

For the virtual experimental data, 100K data on uniform and skewed distribution were generated using the spatial object generator [DaVisual 1.0] as shown in Fig. 5 and Fig. 6. Two distributions were created to evaluate the performance of the multiple compression technique in various environments. The virtual data generation environment variables are shown in Table. 2.

Table 2. Virtual data generation environment

Category	Setting	
Dimensions	2	
Size	(700, 700)	
Boxes created	100000	
High ranges for the box size	(10, 20)	
Variable size of boxes	Yes	
Non-zero size of boxes	Yes	
Seed	8183	25832
Distribution	Skewed	Uniform

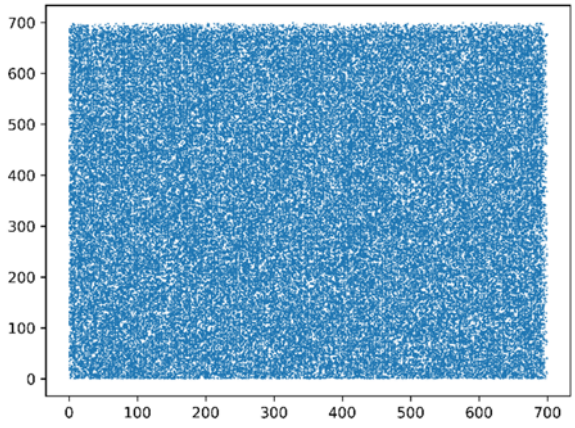

Fig. 5. Simulation Data Set: Uniform MBRs.

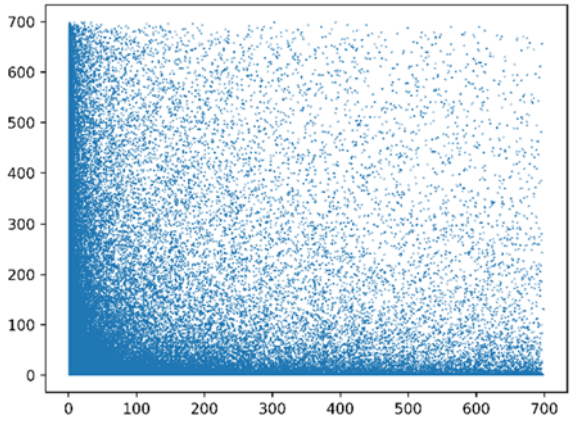

Fig. 6. Simulation Data Set: Skewed MBRs.

Fig. 7 is an algorithm written in Python that compresses into bit string using the grid line of MBR and clustered MBR of spatial data intersecting with x- and y-axis. The algorithm *Multi_compression* receives the dataset *ds* and *bits* for dividing each axis into a grid. Line 1 reads the two diagonal coordinates *(sx, sy)-(ex, ey)* of MBR in *ds* and finds the end coordinates of the x- and y-axis on line 2 because the size of the search area should be identified to split the grid. In lines 3 and 4, two end coordinates, *e_x* and *e_y*, are divided into *bits* to obtain grid units *grids_ux* and *grids_uy* for each axis. In this case, the result is a decimal point, thus the ceil function is used for a wide range. For example, if the maximum coordinate of the region is (3750, 2674) and bits = 8, the grid unit of the x-axis is ceil (3750/8) = 469 and the y-axis is ceil (2674/8) = 335.

The *grid_line_x* and *grid_line_y* of lines 6 and 7 determine the grid lines to expand each coordinate of the MBR to the cell size. Once the grid lines of the x and y coordinates are determined, the bits of the compressed MBR are stored in *zip_x_bits* and *zip_y_bits* and returned as character strings (lines 8–10).

```
Algorithm: Multi_compression (ds, bits)
Input: ds is a dataset in 2-dimension.
       bits is the number of bits
Output: zip_x and zip_y are compressed in bits from x- and y-axes.
01: for (sx, sy, ex, ey) in ds:
02:    (s_x, x_y), (e_x, e_y) from (x, y)
03: grids_ux = ceil (e_x / bits)
04: grids_uy = ceil (e_y / bits)
05: for x, y in ds:
06:    set grid_line_x = x / grids_ux
07:    set grid_line_y = y / grids_uy
08:    zip_x_bits += grid_line_x
09:    zip_y_bits += grid_line_y
10: return (zip_x_bits, zip_y_bits)
```

Fig. 7. Bit-map Transformation Algorithm.

Because the grid of the spatial data or clusters is expressed as MBR in two dimensions, four coordinates of two diagonal points are used. When each coordinate is 4 bytes, a total of 16 bytes of a space is required. Fig. 8 shows the space utilization before and after compression by increasing the storage unit of the hard disk from 512 to 4096 bytes, assuming each coordinate is 4 bytes.

Normal MBRs represent a 16-byte MBR, with 32 stored in a 512-byte space. However, when storing only one MBR in bit compression, it takes 2 bytes and 256 MBRs are stored. In this paper, the multi-compression technique saves more spatial data because three MBRs are stored on one axis. In other words, Multi-Bit Zipped, which represents multi-compression, stores up to 144 MBRs in 4096 storage spaces, which is 24 times more efficient than normal MBRs.

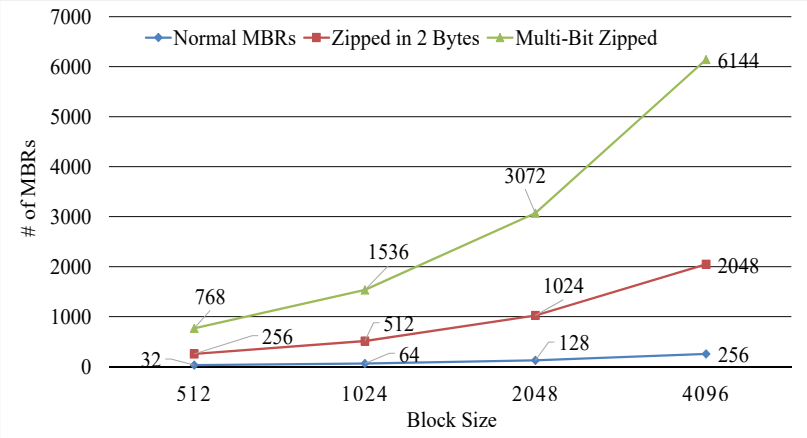

Fig. 8. Block utilization of multi-bit compression.

5. Conclusions

In this study, we proposed a multi-bit compression scheme in machine learning that can improve the storage efficiency and processing speed of clustered data bundles in geographic spatial data. Existing location-based services or clustering techniques have increased the processing efficiency by improving the algorithm, however the performance improvement through spatial data compression was not considered.

Therefore, we divided the search area into grids and extended each side of the MBR that surrounds the spatial data to grid lines. The location where the extended plane meets the grid line was represented by 1 and the location where it did not meet by 0, hence each axis was represented by 8 bits. In particular, the compression rate was increased by storing three MBRs in 8 bits. The experiments showed 24 times better performance in 4K blocks compared to the MBR before compression.

The proposed scheme is expected to increase the service satisfaction of users by providing faster services through coordinate compression in services using spatial data.

Acknowledgments

This paper was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (Ministry of Education) [NRF-2017R1D1A1B03035884].

References

[1] Elghazal, M. S.; Younis, S. A.; Musbah, M. S. (2017): Applying location based services for reducing mobile power consumption. 2017 International Conference on Engineering and Technology (ICET), Antalya, pp. 1–5.
[2] Ekong, P. S.; Cardona, C. J.; Bryssinckx, W.; Ikechukwu-Eneh, C.; Lombin, L. H.; Carpenter, T. E. (2017): Spatial clustering of pathology submissions during the initial introduction and spread of avian influenza H5N1 in poultry in Nigeria in 2006–2007. Veterinaria Italiana, 54(1), pp. 13–20.
[3] Gartner, G.; Huang, H. (2016): Progress in Location-Based Services 2016. Lecture Notes in Geoinformation and Cartography, Springer.
[4] Guttman, A. (1984): R-trees: a dynamic index structure for spatial searching. Proceedings of ACM SIGMOD International Conference on Management of Data, 14, pp. 47–57.
[5] Jannu, S.; Jana, P. K. (2016): A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks. Wireless Networks, Springer, pp. 1901–1916.
[6] Petch, J. (1999): GIS, Organisations and People. CRC Press, 1ed.
[7] Spatial Data Generator, DaVisual Code1.0. Available on: http://isl.cs.unipi.gr.
[8] Wu, B.; Wilamowski, B. M. (2017): A Fast Density and Grid Based Clustering Method for Data With Arbitrary Shapes and Noise. IEEE Transactions on Industrial Informatics, 13(4), pp. 1620–1628.
[9] Zhang, J.; Feng, X.; Liu, Z. (2018): A Grid-Based Clustering Algorithm via Load Analysis for Industrial Internet of Things. IEEE Access, 6, pp. 13117–13128.
[10] Zhou, E.; Mao, S,; Li, M.; Sun, Z. (2016): PAM spatial clustering algorithm research based on CUDA. 2016 24th International Conference on Geoinformatics, Galway, pp. 1 –7.