An Extension of the Automatic Cross-Association Method with a 3-dimensional Matrix

Won-Jo Lee[1], Chae-Gyun Lim[2], U Kang[3], and Ho-Jin Choi[4]
Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
{mochagold[1], rayote[2], hojinc[4]}@kaist.ac.kr, ukang@cs.kaist.ac.kr[3]

***Abstract*— There are numerous 2-dimensional matrix data for clustering including a set of documents, citation networks, web graphs, etc. However, many real-world datasets have more than three modes which require at least 3-dimensional matrices or tensors. Focusing on the clustering algorithm known as cross-association, we extend the algorithm to deal with a 3-dimensional matrix. Our proposed method is fully automated, and simultaneously discovers clusters of both row, column, and tube groups. Experiments on real and synthetic datasets show that our method is effective. Through the proposed method, useful information can be obtained even from sparse datasets.**

Keywords—3-dimensional matrix; clustering; cross association; data analysis

I. Introduction

Recently, many studies for finding groups and patterns using clustering or classification methods have been reported [1][2]. Techniques for finding specific patterns or information from scattered and chaotic data are useful and can be used for a number of domains.

Most clustering algorithms for matrices focus on 2-dimensional matrix data. For example, analysis of a market basket involves 2-dimentional matrix data [3][4]; document analysis involves a data matrix containing a set of documents and a bag of words [7][8]; a graph is often represented by a 2-dimensional matrix [10][11]. However, in the real world, there are many matrices or tensors having at least 3 dimensions: knowledge bases containing triples of subject, object, and verb [12]; lymphoma microarray dataset in biology analysis [13][5][6]; DBLP citation network containing author, keyword, and conference; app recommendation data containing user, app, and location entities [14]; and movie recommendation data containing user, movie, and time [15]. However, to the best of our knowledge, most clustering algorithms targeting over 3-dimensional matrix are based on tensor factorization [16][17] or k-means clustering.

In this paper, we focus on cross-association [18] due to a number of advantages. Cross-association is based on co-clustering, and therefore is able to simultaneously discover both row and column groups. Moreover, cross-association is fully automated, meaning that it does not require any parameters, e.g., the number of clusters. However, cross-association has limitations: it only targets sparse binary 2-dimentional matrices. Therefore, in this paper, we extend cross-association to handle 3-dimensonal matrix. The proposed method simultaneously discovers clusters in a 3-dimensional matrix and is fully automatic, like cross-association. Using the proposed method, we cluster and analyze 3-dimensonal real-world matrix data.

II. Related work

In this section, we discuss several algorithms for finding patterns, correlations, and rules through clustering. They include spectral clustering [10], METIS [11], co-clustering [7], and cross-association [18]. Except cross-association, all algorithms we introduce require tuning and human intervention.

Spectral clustering is based on finding “good cuts” in a graph. It finds clusters which contain dense internal edges and sparse external edges. Several versions of spectral clustering have been proposed, including the one using the ratio cut [9] and the one using the normalized cut [10]. METIS provides multilevel k-way partitioning. The result of METIS is a set of balanced clusters of graph nodes, where the number of edges between clusters is small. Co-clustering is able to find clusters of rows and columns from given matrices or graphs simultaneously. It normalizes a contingency table or a two-dimensional probability distribution. Cross-association is from the co-clustering method. It simultaneously groups the row items and the column items from a given similarity matrix. The difference between Information Co-Clustering [8] and cross-association is that Information Co-Clustering uses lossy code, which is for rank-one matrix approximation, while Cross-association takes lossless code, which uses the Shannon entropy function. One of the important properties of cross-association is that it is fully automated, meaning that it does not require any parameters. However, it targets only sparse binary 2-dimensional matrices.

III. Proposed Method

The goal of this paper is to find clusters in 3-dimensional matrix data by extending Cross-association. The main idea of cross-association is to exploit the Minimum Description Length [20] principle: it finds clusters which minimize the number of bits required to transmit both the summary of the structure, as well as each rectangular area. Table 1 shows the symbols and definitions used in this paper.

A. Extended Cross-association

Compared to 2-D cross association, the key difference of our method is that it targets a 3-dimensional matrix. Let D denote an α x β x γ $(\alpha, \beta, \gamma \geq 1)$ matrix. Let us index the rows, columns, and tubes as $(1, 2, \dots, \alpha)$, $(1, 2, \dots, \beta)$, and $(1,2, \dots, \gamma)$,

TABLE I. SYMBOLS AND DEFINITIONS.

Symbol	Definition
D α, β, γ	3-dimentional matrix Number of row, columns, and tubes in D
$\mathrm{k}, \mathrm{l}, \mathrm{m}$ $\mathrm{k}^*, \mathrm{l}^*, \mathrm{m}^*$ (Φ, Ψ, Ω)	Number of rows, columns, and tube groups Optimal numbers of groups Extended Cross-association
$\mathrm{D}_{\mathrm{i,j,h}}$ $\mathrm{a}_\mathrm{i}, \mathrm{b}_\mathrm{j}, \mathrm{c}_\mathrm{h}$ $n(\mathrm{D}_{\mathrm{i,j,h}})$ $\mathrm{n}_0(D_{i,j,h}), n_1(D_{i,j,h})$	(i,j,h)th cross-associate Dimensions of $\mathrm{D}_{\mathrm{i,j,h}}$ Number of elements in $\mathrm{D}_{\mathrm{i,j,h}}$ Number of 0s and 1s in $\mathrm{D}_{\mathrm{i,j,h}}$
$\mathrm{H(p)}$	Shannon entropy function
$\mathrm{C}(\mathrm{D}_{\mathrm{i,j,h}})$ $\mathrm{T}(\mathrm{D}; \mathrm{k}, \mathrm{l}, \mathrm{m}, \Phi, \Psi, \Omega)$	Code cost for $\mathrm{D}_{\mathrm{i,j,h}}$ Total cost for D

respectively. Let k, l, and m denote the expected number of disjoint row groups, column groups, and tube groups, respectively. And let us index the row groups as $(1, 2, \ldots, \mathrm{k})$, column groups as $(1, 2, \ldots, \mathrm{l})$ and tube groups as $(1,2, \ldots, \mathrm{m})$. Let $\Psi: \{1, 2, \ldots, \alpha\} \rightarrow \{1, 2, .., k\}, \Phi: \{1, 2, \ldots, \beta\} \rightarrow \{1, 2, \ldots, l\}$, and $\Omega: \{1, 2, \ldots \gamma\} \rightarrow \{1, 2, .., \mathrm{m}\}$ denote the results of assignments of rows to row groups, columns to column groups, and tubes to tube groups. Extended Cross-association outputs Ψ, Φ, and Ω. With the given Extended Cross-association, we are able to gain more insights on the structure of D. Using Ψ, we rearrange rows of D so that rows corresponding to group 1 is listed first, rows corresponding to group 2 is listed second, and so on. Columns and tubes are rearranged in a similar manner based on Φ and Ω. Through this rearrangement, matrix D is divided into smaller rectangular blocks. We denote each cross-associate as $\mathrm{D}_{\mathrm{i,j,h}}$, where $i = 1,2, \ldots, k,\ j = 1,2, \ldots, l$, and $h = 1,2, \ldots, m$. The dimensions of $\mathrm{D}_{\mathrm{i,j,h}}$ are denoted by $(\mathrm{a}_\mathrm{i}, \mathrm{b}_\mathrm{j}, \mathrm{c}_\mathrm{h})$.

B. Compression

The total code length for the given matrix D, with respect to a given Extended Cross-association, comprises two parts: the description (model) complexity and the data complexity. Description complexity considers the cost of storing the following model parameters:

- Number of row groups, column groups, and tube groups,
- Number of rows in each row group, number of columns in each column group, number of tubes in each tube group
- Number of ones in each cross-associate $\mathrm{D}_{\mathrm{i,j,h}}$

Let A be an $\mathrm{a} \times \mathrm{b} \times \mathrm{c}$ matrix. Let $n_1(A)$ and $n_0(A)$ be the number of non-zero entries and zero entries in A. The optimal code length to encode A is given by [18], as in (1):

$$\mathrm{C(A)} \coloneqq \sum_{i=0}^{1} n_i(A) \log\left(\frac{n(A)}{n_i(A)}\right) = n(A) H(P_A(0)) \qquad (1)$$

where H() is the Shannon entropy function. The data complexity considers the number of bits to encode D using an optimal code. For the given matrix D, the total code length is as in (2).

$$\mathrm{T}(\mathrm{D}; \mathrm{k}, \mathrm{l}, \mathrm{m}, \Phi, \Psi, \Omega) \coloneqq \log^* k + \log^* l + \log^* m +$$

$$\sum_{i}^{k-1} \lceil \log \overline{a_i} \rceil + \sum_{j}^{l-1} \lceil \log \overline{b_j} \rceil + \sum_{h}^{m-1} \lceil \log \overline{c_h} \rceil +$$

$$\sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{h=1}^{m} \lceil \log(a_i b_j c_h) + 1 \rceil + \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{h=1}^{m} C(D_{i,j,h}) \qquad (2)$$

where $\overline{a_i}$ is defined to be $(\sum_{t=i}^{k} a_t) - k + i$. $\overline{b_j}$ and $\overline{c_h}$ are defined in a similar manner. Our goal is to find the number of row groups k^*, the number of column groups l^*, and the number of tube groups m^* , and an Extended Cross-association $(\Psi^*, \Phi^*, \Omega^*)$ such that the total resulting code length $\mathrm{T}(\mathrm{D}; \mathrm{k}^*, \mathrm{l}^*, \mathrm{m}^*, \Psi^*, \Phi^*, \Omega^*)$ is minimized. To determine the proper number of groups in each dimension, and also a corresponding Extended Cross-association, we propose a two-step approach. The first step is for finding a good Extended Cross-association for a given number of rows, columns, and tube groups. The second part is for finding the proper number of rows, columns, and tube groups. In the following, we first explain the strategy to search over k, l, and m to minimize the total code length T mentioned above. Then, a minimization algorithm called *Extended-ReGroup* to find an optimal Extended Cross-association for a given number of groups in each dimension is then proposed.

C. Extended Cross-association Search

In this part, we propose an algorithm to find the proper number of groups, k, l, and m. It is based on the total cost model. Staring with the minimum number ($\mathrm{k}^0 = \mathrm{l}^0 = \mathrm{m}^0 = 1$), we increase each of the number of groups. For each increase, the matrix is then rearranged with *Extended-ReGroup* which will be described at the next section. In contrast to the 2-D cross association which alternates increasing the number of rows, and columns, resp., we choose the best dimension (row or column or tube) that decreases the total cost the most.

D. Extended Regroup

Extended ReGroup finds the local minimum of Extended Cross-association. If we have the number of row groups k, the number of column groups l, and the number of tube groups m, we focus on obtaining an Extended Cross-association $(\Psi^*, \Phi^*, \Omega^*)$ that minimizes the cost function.

$$\sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{h=1}^{m} C(D_{i,j,h}) \qquad (3)$$

The idea is alternating minimization, i.e., alternately choose a dimension, and for each index of the dimension, choose the best group assignment that minimizes the cost function.

IV. EXPERIMENTS

To validate the usefulness and clustering performance of our method, we carried out experiments with two kinds of data, synthetic and real-world. The synthetic dataset includes a block-diagonal 3-dimensional matrix with varied block sizes, and its shuffled version. The real-world dataset includes the DBLP (ver. 7) citation network data [21] from AMiner containing 2,244,021 papers and 4,354,534 citation relationships.

With the synthetic data, we show that our method catches clusters in a 3-D matrix by rearranging entity indexes. Figure 1 shows a block-diagonal raw data, and Figure 2 shows the shuffled version of the data in Figure 1. We apply our method to find clusters in the data in Figure 2. The resulting 3-D matrix is exactly the same as the data in Figure 1, as expected. Figure 3 shows the resulting 3-D matrix, where the *i*th subfigure is a collapsed version (along the z-axis) of the 3-D submatrix whose group assignment in the z-axis is *i*. The darkness of a box represents the density of it.

In the real-world data experiment, we use the DBLP (paper, author, year) 3-D matrix data as shown in Figure 4. The number of papers, authors, and years is 3000, 5000, and 50, respectively. We apply our method on the data, and as a result we found 4 paper groups, 3 author groups, and 4 year groups. The size of the paper groups, author groups, and year groups is {1424, 803, 412, 362}, {3076, 568, 665}, and {30, 9, 1, 1}, respectively. Figure 5 shows the resulting 3-D matrix, where the *i*th subfigure is a collapsed version (along the z-axis) of the 3-D sub matrix whose group assignment in the z-axis is *i*. As in the synthetic data experiment, the darkness of a box represents the density of it. We see that each subfigure has a small number of dense blocks, rather than many sparse blocks, as intended by our method.

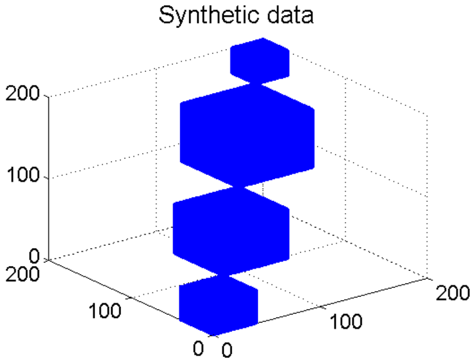

Fig. 1. Block diagonal 3-D matrix. X-axis, y-axis, and z-axis has ranges from 0 to 199, respectively.

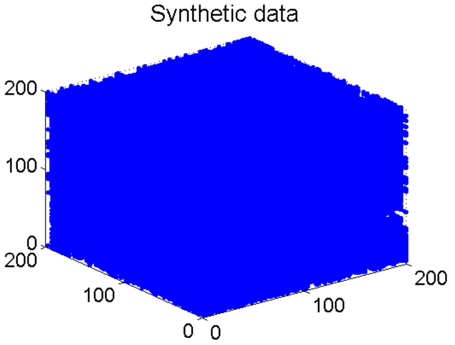

Fig. 2. Shuffled data from Fig. 1.

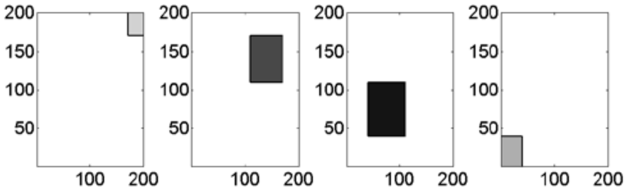

Fig. 3. The resulting 3-D matrix after applying our method to the data in Fig. 2. The *i*th subfigure is a collapsed version (along the z-axis) of the 3-D sub matrix whose group assignment in the z-axis is *i*.

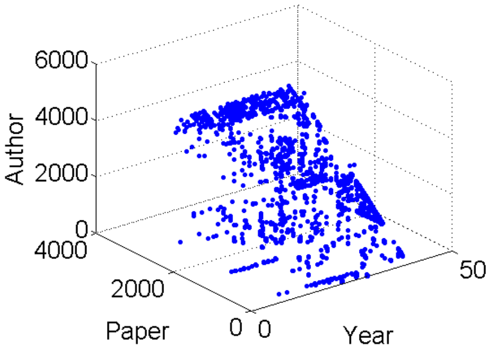

Fig. 4. Visualization of DBLP dataset. Paper index is from 1 to 3000, author index is from 1 to 5000, and year index is from 1 to 50.

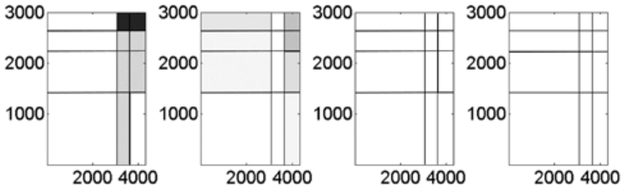

Fig. 5. The resulting 3-D matrix after applying our method to the data in Fig. 4. The *i*th subfigure is a collapsed version (along the z-axis) of the 3-D submatrix whose group assignment in the z-axis is *i*. The darkness of a box represents the density of it. Note that each subfigure has a small number of dense blocks, rather than many sparse blocks, as intended by our method.

V. Conclusion

In this paper, we proposed *Extended Cross-association*, an algorithm that extends 2-D cross-association to handle 3-dimensional matrices or tensors. Our proposed method is parameter-free, and it simultaneously finds row, column, and tube groups. Experiments on real world DBLP data and synthetic dataset show our method is effective.

Future research topics include 1) more experiments with real world data, and 2) improving the performance of Extended Cross-association to handle much larger data.

Acknowledgment

This work was supported by the Korea Ministry of Science, ICT and Future Planning, under the title “Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services” (No. 10044577), and through the National Research Foundation of Korea (No. 2013R1A1A1064409).

References

[1] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[2] M. EJ Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, pp. 026113, 2004.

[3] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” *in VLDB*, vol. 1215, pp. 487–499, 1994.

[4] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, “Dynamic itemset counting and implication rules for market basket data,” *in ACM SIGMOD Record*, vol. 26, no. 2, pp. 255-265, 1997.

[5] U. Kang, E. E. Papalexakis, A. Harpale, and C. Faloutsos, “Gigatensor: scaling tensor analysis up by 100 times - algorithms and discoveries,” *in KDD*, pp. 316–324, 2012

[6] E. E. Papalexakis, U. Kang, C. Faloutsos, N. D. Sidiropoulos, and A. Harpale, “Large scale tensor decompositions: Algorithmic developments and applications,” *IEEE Data Engineering Bulletin Issues*, vol. 36, no. 3, pp. 59–66, Sept. 2013

[7] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” *in KDD*, pp. 269-274, 2001.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering,” *in KDD,* pp. 89-98, 2003.

[9] S. Wang and J. Mark Siskind, “Image segmentation with ratio cut,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675-690, 2003.

[10] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888–905, 2000.

[11] G. Karypis and V. Kumar, “Multilevel k-way partitioning scheme for irregular graphs,” *Parallel and Distributed computing*, vol. 48, no. 1, pp. 96–129, 1998.

[12] T. Van de Cruys, “A non-negative tensor factorization model for selectional preference induction,” *Natural Language Engineering*, vol. 16, issue 4, pp. 417-437, 2010.

[13] S. Jegelka, S. Sra, and A. Banerjee, “Approximation algorithms for tensor Clustering,” *Algorithmic learning theory. Springer Berlin Heidelberg*, pp. 368-383, 2009.

[14] A. Karatzoglou, L. Baltrunas, K. Church, and M. Bohmer, “Climbing the app wall: enabling mobile app discovery through context-aware recommendations,” *in CIKM*, pp. 2527-2530, 2012.

[15] K. Shin, and U. Kang, "Distributed Methods for High-dimensional and Large-scale Tensor Factorization," *in ICDM*, 2014.

[16] U. Kang, "Mining Tera-Scale Graphs: Theory, engineering and discoveries,", Ph.D. thesis, Carnegie Mellon University, 2012

[17] I. Jeon, E. E. Papalexakis, U Kang, and C. Faloutsos, "HaTen2: Billion-scale Tensor Decompositions," *in ICDE*, 2015

[18] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos, “Fully automatic cross-associations,” *in KDD*, pp. 79-88, 2004.

[19] S. Wang and J. Mark Siskind, “Image segmentation with ratio cut,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675-690, 2003.

[20] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465-471, 1978

[21] (2014) Citation Network dataset [Online]. Avaliable: http://arnetminer.org/citation