Star-Structured High-Order Heterogeneous Data Co-clustering based on Consistent Information Theory

Bin Gao Tie-Yan Liu Wei-Ying Ma
Microsoft Research Asia
4F, Sigma Center, No. 49, Zhichun Road
Beijing, 100080, P. R. China
{bingao, tyliu, wyma}@microsoft.com

Abstract

Heterogeneous object co-clustering has become an important research topic in data mining. In early years of this research, people mainly worked on two types of heterogeneous data (denoted by pair-wise co-clustering); while recently more and more attention was paid to multiple types of heterogeneous data (denoted by high-order co-clustering). In this paper, we studied the high-order co-clustering of objects with star-structured inter-relationship, i.e., there is a central type of objects that connects the other types of objects. Actually, this case could be a very good model for many real-world applications, such as the co-clustering of Web images, their low-level visual features, and the surrounding text. We used a tripartite graph to represent the inter-relationships among different objects, and proposed a consistent information theory which generates an effective algorithm to obtain the co-clusters of different types of objects. Experiments on a Web image show that our proposed algorithm is a better choice compared with previous work on heterogeneous object co-clustering.

1. Introduction

Homogeneous data clustering has been extensively studied in the literature. However, in real applications nowadays, people often need to deal with the co-clustering of heterogeneous data objects. For example, in order to conduct personalized search, one may need to co-cluster three types of objects in the click-through log of Web search engines, i.e., Web users, issued queries, and clicked pages. One may also need to co-cluster Web images, their low-level visual features, and surrounding text for a friendly user interface of an image search engine. And one may need to co-cluster papers, authors, conferences, and journals in order to mine communities in the research society. In the above examples, not only features of the objects, but also their inter-relationships are heterogeneous. In such cases, homogeneous clustering technologies cannot work well any longer.

In the past decades, researchers have proposed many technologies on the co-clustering of two types of heterogeneous data (denoted by pair-wise co-clustering). Representative work includes information bottleneck co-clustering [6][10], bipartite spectral graph partitioning [4], and information theoretic co-clustering [1][5]. According to [5], the information theoretic method is much more scalable and efficient than the other methods. In this particular method, a two-dimensional co-occurrence table, which describes the relations among data objects, was viewed as an empirical joint probability distribution of two discrete random variables. Accordingly, the co-clustering problem was posed as an optimization problem in information theory: the optimal co-clustering maximized the mutual information [3] between the clustered random variables subject to constraints on the number of row and column clusters in the co-occurrence table. Besides the above work on pair-wise co-clustering, researchers also made some efforts on the co-clustering of multiple types of objects (denoted by high-order co-clustering). For example, Wang *et al* [11] proposed an iterative method named ReCoM to cluster multi-type interrelated Web objects.

In this paper, we would like to study a special topology of star-structured high-order heterogeneous data as shown in Figure 1.(a). In this structure, there is a central type of objects that connects the other types of objects. This case could be a very good abstraction for many real-world applications, such as the co-clustering of authors, conferences, papers, and keywords in academic publication systems (corresponding to Figure 1.(a).(ii), where paper is the central data type), in order to identify that a certain group of authors usually write papers of a certain series of topics using a certain list of keywords, and submit them to a certain kind of conferences. To solve the problem of star-structured high-order co-clustering, we designed an algorithm named consistent bipartite graph co-partitioning (CBGC) based on the *consistency theory* proposed in [7][8]. Although experiments showed this algorithm is very effective, it is not very efficient in large-scale datasets because CBGC is solved by semi-definite programming (SDP), which is time-consuming in large-scale cases.

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00

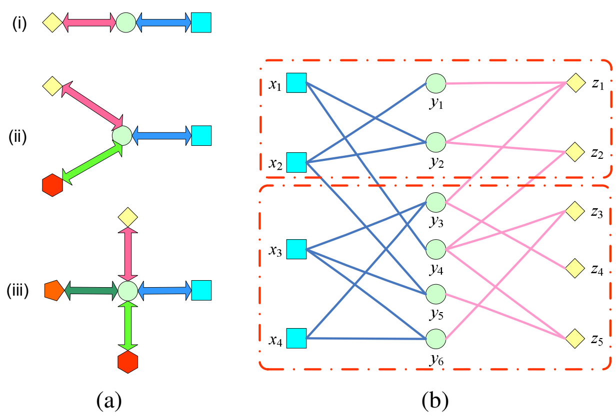

Figure 1. (a) Star-structured high-order heterogeneous data. (b) The tripartite graph of heterogeneous objects.

The motivation of this paper is to develop a more efficient method to solve the aforementioned star-structured high-order co-clustering problem. For simplicity, we only focus on the case as shown in Figure 1.(a).(i). In order to get a high-order co-clustering algorithm with both high effectiveness and speed, we extend the information theoretic co-clustering algorithm [5] to the high-order case, once again based on the consistency theory. Similar to [7][8], we use a tripartite graph to represent the inter-relationships among different types of objects. Figure 1.(b) shows a tripartite graph which consists of three types of heterogeneous objects: $X=\{x_1, x_2, \ldots, x_m\}$, $Y=\{y_1, y_2, \ldots, y_n\}$, and $Z=\{z_1, z_2, \ldots, z_l\}$. We model the co-clustering of X, Y, and Z as the consistent fusion of two pair-wise co-clustering sub-problems. That is, we look for such two partitions for the sub-problems of X-Y co-clustering and Z-Y co-clustering, provided that their clustering results on the central type Y are the same. Then for each sub-problem, we adopt the information theoretic co-clustering algorithm to get the desirable clusters, and adjust these clusters by considering the clustering results of the other sub-problem. Experiments on a Web image dataset show that this method is almost as effective as the CBGC method, while it is much faster. In this regard, this method is more suitable for real-world large-scale co-clustering applications.

2. Problem formulation

In the following two sections, we will mainly discuss how to extend the information-theoretic co-clustering method to handle the star-structured high-order cases. First of all, we will introduce a probability model to represent the inter-relationship among heterogeneous objects, which is an extension of that used in [4].

Let X, Y, and Z be discrete random variables that take values from the sets $\{x_1, \ldots, x_m\}$, $\{y_1, \ldots, y_n\}$, and $\{z_1, \ldots, z_l\}$, representing the three types of objects. Denote the joint probability distributions between X and Y, and between Z and Y as the $m \times n$ matrix $p_1(X,Y)$ and the $l \times n$ matrix $p_2(Z,Y)$. Our target is to cluster X, Y, and Z into r, s, and t disjoint (or hard) clusters simultaneously. Suppose the clusters of X, Y, and Z are $\{\hat{x}_1,\ldots,\hat{x}_r\}$, $\{\hat{y}_1,\ldots,\hat{y}_s\}$, and $\{\hat{z}_1,\ldots,\hat{z}_t\}$, we are actually seeking the maps C_X, C_Y, and C_Z, i.e., *(i)* $C_X:\{x_1,\ldots,x_m\}\rightarrow\{\hat{x}_1,\ldots,\hat{x}_r\}$, *(ii)* $C_Y:\{y_1,\ldots,y_n\}\rightarrow\{\hat{y}_1,\ldots,\hat{y}_s\}$, *(iii)* $C_Z:\{z_1,\ldots,z_l\}\rightarrow\{\hat{z}_1,\ldots,\hat{z}_t\}$. In brief, we denote $\hat{X}=C_X(X)$, $\hat{Y}=C_Y(Y)$, and $\hat{Z}=C_Z(Z)$. Based on the above notations, we have the following definitions.

Definition 1. *We refer to the tuple (C_X, C_Y) as a co-clustering.*

Definition 2. *We refer to the star-structured triple (C_X, C_Y, C_Z) as a consistent co-clustering, where C_Y is the mapping corresponding to the central data type.*

A traditional and fundamental quantity that measures the amount of information that random variable X contains about Y (and *vice versa*) is the mutual information $I(X,Y)$ [3]. As shown in Definition 3, similar to [5], the resultant loss in mutual information is adopted in this paper to judge the quality of a co-clustering.

Definition 3. *An optimal co-clustering minimizes $I(X,Y)$ $-I(\hat{X},\hat{Y})$ subject to the constraints on the number of row and column clusters in the probability matrix $p_1(X,Y)$, where $I(\cdot,\cdot)$ denotes the mutual information, i.e.,*

$$I(X,Y)=\sum_x\sum_y p_1(x,y)\log\left(p_1(x,y)/(p_1(x)p_1(y))\right). \quad (1)$$

Similarly, we have

$$I(Z,Y)=\sum_z\sum_y p_2(z,y)\log\left(p_2(z,y)/(p_2(z)p_2(y))\right). \quad (2)$$

It was proved in [5] that the loss of mutual information could be obtained by calculating a Kullback-Leibler (KL) [3] divergence. For the star-structured high-order co-clustering, we can have the similar conclusion as follows after some deductions.

Lemma 1. *For a fixed consistent co-clustering (C_X, C_Y, C_Z), we can write the loss in mutual information as*

$$I(X,Y)-I(\hat{X},\hat{Y})=D(p_1(X,Y)\,\|\,q_1(X,Y)), \quad (3)$$

$$I(Z,Y)-I(\hat{Z},\hat{Y})=D(p_2(Z,Y)\,\|\,q_2(Z,Y)), \quad (4)$$

where $D(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence, also known as relative entropy, and $q_1(X,Y)$ and $q_2(Z,Y)$ are distributions of the following forms:

$$q_1(x,y)=p_1(\hat{x},\hat{y})p_1(x\,|\,\hat{x})p_1(y\,|\,\hat{y}),\ \text{where } x\in\hat{x},\ y\in\hat{y}, \quad (5)$$

$$q_2(z,y)=p_2(\hat{z},\hat{y})p_2(z\,|\,\hat{z})p_2(y\,|\,\hat{y}),\ \text{where } z\in\hat{z},\ y\in\hat{y}. \quad (6)$$

According to the consistency theory [7][8], we divide the original X-Y-Z co-clustering problem into two sub-problems: X-Y co-clustering and Z-Y co-clustering, with the constraints that their clustering results on the central type Y are exactly the same and the overall partitioning is optimal under a certain objective function. A simple but

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

feasible objective function could just be the linear combination of the two KL divergences in (3) and (4).

Definition 4. *The objective function of the consistent co-clustering on (C_X, C_Y, C_Z) is defined as*

$$F(X,Y,Z) = \alpha D(p_1(X,Y) \| q_1(X,Y)) + (1-\alpha) D(p_2(Z,Y) \| q_2(Z,Y)), \text{ where } 0 < \alpha < 1. \quad (7)$$

In the right-hand side of the above formula, the first term stands for the objective function for the sub-problem of *X-Y* co-clustering, while the second one stands for that of *Y-Z* co-clustering. Parameter α is a weighting factor determining which local bipartite graph we trust more.

3. Consistent information theoretic co-clustering

According to [5], we have the following proposition and lemma after some extensions.

Proposition 1. *For the sub-problems of X-Y co-clustering and Z-Y co-clustering in a fixed (or hard) consistent co-clustering (C_X, C_Y, C_Z), there holds*

$$D(p_1(X,Y) \| q_1(X,Y)) = D(p_1(X,Y,\hat{X},\hat{Y}) \| q_1(X,Y,\hat{X},\hat{Y})), \quad (8)$$

$$D(p_2(Z,Y) \| q_2(Z,Y)) = D(p_2(Z,Y,\hat{Z},\hat{Y}) \| q_2(Z,Y,\hat{Z},\hat{Y})). \quad (9)$$

Lemma 2. *For either of the sub-problems of X-Y co-clustering and Z-Y co-clustering in a fixed (or hard) consistent co-clustering (C_X, C_Y, C_Z), the loss in mutual information can be expressed as (i) a weighted sum of the relative entropies between row distributions and "row-lumped" distributions, or as (ii) a weighted sum of the relative entropies between column distributions and "column-lumped" distributions, that is,*

$$D(p_1(X,Y,\hat{X},\hat{Y}) \| q_1(X,Y,\hat{X},\hat{Y})) = \sum_{\hat{x}} \sum_{x:C_X(x)=\hat{x}} p_1(x) D(p_1(Y \mid x) \| q_1(Y \mid \hat{x})), \quad (10)$$

$$D(p_1(X,Y,\hat{X},\hat{Y}) \| q_1(X,Y,\hat{X},\hat{Y})) = \sum_{\hat{y}} \sum_{y:C_Y(y)=\hat{y}} p_1(y) D(p_1(X \mid y) \| q_1(X \mid \hat{y})). \quad (11)$$

$$D(p_2(Z,Y,\hat{Z},\hat{Y}) \| q_2(Z,Y,\hat{Z},\hat{Y})) = \sum_{\hat{z}} \sum_{z:C_Z(z)=\hat{z}} p_2(z) D(p_2(Y \mid z) \| q_2(Y \mid \hat{z})), \quad (12)$$

$$D(p_2(Z,Y,\hat{Z},\hat{Y}) \| q_2(Z,Y,\hat{Z},\hat{Y})) = \sum_{\hat{y}} \sum_{y:C_Y(y)=\hat{y}} p_2(y) D(p_2(Z \mid y) \| q_2(Z \mid \hat{y})). \quad (13)$$

Lemma 2 shows that in either of the sub-problems, we can express the objective function solely in terms of the row-clustering, or in terms of the column-clustering. Then, for example, in *X-Y* co-clustering, we can define the distribution $q_1(Y \mid \hat{x})$ as a *row-cluster prototype*, and similarly, the distribution $q_1(X \mid \hat{y})$ as a *column-cluster prototype*. Based on this intuition, the co-clustering of each sub-problem can be calculated by iteratively computing row clusters and column clusters. Furthermore, it can be proved that this interactive process can gradually maximize the mutual information between the clustered random variables in a reinforcing manner. To solve the overall consistent co-clustering problem, we first use the above idea to get the solution of each sub-problem, and then determine the clustering result for the central type of object *Y* by minimizing the objective function as defined in Definition 4, based on the foregoing clustering results of *X* and *Z*. This process can be conducted in an iterative way, until the co-clustering results become stable. More specifically, we propose the **C**onsistent **I**nformation **T**heoretic co-clustering algorithm (**CIT**) as shown in Table 1 to solve the star-structured high-order heterogeneous data co-clustering problem.

Table 1. The CIT algorithm.

ALGORITHM CIT (p_1, p_2, r, s, t, α, &C_X, &C_Y, &C_Z)

Input: p_1: the joint probability distributions of X and Y; p_2: the joint probability distributions of Z and Y; r: the desired cluster number of X; s: the desired cluster number of Y; t: the desired cluster number of Z.

Output: The mapping functions C_X, C_Y and C_Z.

1. Initialization: Set i=0. Start with some initial partition functions $C_X^{(0)}, C_Y^{(0)}$ and $C_Z^{(0)}$. Compute

   $$q_1^{(0)}(\hat{X},\hat{Y}), q_1^{(0)}(X \mid \hat{X}), q_1^{(0)}(Y \mid \hat{Y}), q_2^{(0)}(\hat{Z},\hat{Y}), q_2^{(0)}(Z \mid \hat{Z}), q_2^{(0)}(Y \mid \hat{Y}),$$

   and distributions $q_1^{(0)}(Y \mid \hat{x}), 1 \le \hat{x} \le r$ and $q_2^{(0)}(Y \mid \hat{z}), 1 \le \hat{z} \le t$ using $q_1^{(i)}(y \mid \hat{x}) = q_1^{(i)}(y \mid \hat{y}) q_1^{(i)}(\hat{y} \mid \hat{x})$, $q_2^{(i)}(y \mid \hat{z}) = q_2^{(i)}(y \mid \hat{y}) q_2^{(i)}(\hat{y} \mid \hat{z})$.

2. Compute X clusters. For each x, find its new cluster index as

   $$C_X^{(i+1)}(x) = \arg\min_{\hat{x}} D(p_1(Y \mid x) \| q_1^{(i)}(Y \mid \hat{x})),$$

   resolving ties arbitrarily. Let $C_Y^{(i+1)} = C_Y^{(i)}$.

3. Compute distributions $q_1^{(i+1)}(\hat{X},\hat{Y}), q_1^{(i+1)}(X \mid \hat{X}), q_1^{(i+1)}(Y \mid \hat{Y})$

   and the distributions $q_1^{(i+1)}(X \mid \hat{y}), 1 \le \hat{y} \le s$ using

   $$q_1^{(i+1)}(x \mid \hat{y}) = q_1^{(i+1)}(x \mid \hat{x}) q_1^{(i+1)}(\hat{x} \mid \hat{y}).$$

4. Compute Y clusters. For each y, find its new cluster index as

   $$C_Y^{(i+2)}(y) = \arg\min_{\hat{y}} p_1(X \mid y) \| q_1^{(i+1)}(X \mid \hat{y})$$

   resolving ties arbitrarily. Let $C_X^{(i+2)} = C_X^{(i+1)}$.

5. Compute distributions $q_1^{(i+2)}(\hat{X},\hat{Y}), q_1^{(i+2)}(X \mid \hat{X}), q_1^{(i+2)}(Y \mid \hat{Y})$

   and the distributions $q_1^{(i+2)}(Y \mid \hat{x}), 1 \le \hat{x} \le r$ using

   $$q_1^{(i)}(y \mid \hat{x}) = q_1^{(i)}(y \mid \hat{y}) q_1^{(i)}(\hat{y} \mid \hat{x}).$$

6. If the number of the process loop of Steps 2~5 exceeds the scheduled value, or the change in objective function value of the *X-Y* sub-problem, that is,

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

$$D(p_1(X,Y)\,||\,q_1^{(i)}(X,Y)) - D(p_1(X,Y)\,||\,q_1^{(i+2)}(X,Y)),$$

is small, go to Step 7; otherwise, go to Step 2.

7. Compute Z clusters. For each z, find its new cluster index as

$$C_Z^{(i+1)}(z) = \arg\min_{\hat{z}} D(p_2(Y \mid z)\,||\,q_2^{(i)}(Y \mid \hat{z})),$$

resolving ties arbitrarily. Let $C_Y^{(i+1)} = C_Y^{(i)}$.

8. Compute distributions $q_2^{(i+1)}(\hat{Z},\hat{Y}),\, q_2^{(i+1)}(Z \mid \hat{Z}), q_2^{(i+1)}(Y \mid \hat{Y})$

and the distributions $q_2^{(i+1)}(Z \mid \hat{y}), 1 \le \hat{y} \le n$ using

$$q_2^{(i+1)}(z \mid \hat{y}) = q_2^{(i+1)}(z \mid \hat{z}) q_2^{(i+1)}(\hat{z} \mid \hat{y}).$$

9. Compute Y clusters. For each y, find its new cluster index as

$$C_Y^{(i+2)}(y) = \arg\min_{\hat{y}} D(p_2(Z \mid y)\,||\,q_2^{(i+1)}(Z \mid \hat{y}))$$

resolving ties arbitrarily. Let $C_Z^{(i+2)} = C_Z^{(i+1)}$.

10. Compute distributions $q_2^{(i+2)}(\hat{Z},\hat{Y}),\, q_2^{(i+2)}(Z \mid \hat{Z}), q_2^{(i+2)}(Y \mid \hat{Y})$

and the distributions $q_2^{(i+2)}(Y \mid \hat{z}), 1 \le \hat{z} \le t$ using

$$q_2^{(i)}(y \mid \hat{z}) = q_2^{(i)}(y \mid \hat{y}) q_2^{(i)}(\hat{y} \mid \hat{z}).$$

11. If the number of the process loop of Steps 7~10 exceeds the scheduled value, or the change in objective function value of the Z-Y sub-problem, that is,

$$D(p_2(Z,Y)\,||\,q_2^{(i)}(Z,Y)) - D(p_2(Z,Y)\,||\,q_2^{(i+2)}(Z,Y)),$$

is small, go to Step 12; otherwise, go to Step 7.

12. Compute Y clusters under the concept of consistency. For each y, find its new cluster index as

$$\begin{aligned} C_Y^{(i+2)}(y) = \arg\min_{\hat{y}} [&\alpha p_1(y) D(p_1(X \mid y)\,||\,q_1^{(i+1)}(X \mid \hat{y})) \\ &+ (1-\alpha) p_2(y) D(p_2(Z \mid y)\,||\,q_2^{(i+1)}(Z \mid \hat{y}))] \end{aligned} \quad (14)$$

resolving ties arbitrarily ($0 < \alpha < 1$). Let $C_X^{(i+2)} = C_X^{(i+1)}$ and $C_Z^{(i+2)} = C_Z^{(i+1)}$.

13. Compute distributions

$$q_1^{(i+2)}(\hat{X},\hat{Y}), q_1^{(i+2)}(X \mid \hat{X}), q_1^{(i+2)}(Y \mid \hat{Y}),\ q_2^{(i+2)}(\hat{Z},\hat{Y}), q_2^{(i+2)}(Z \mid \hat{Z}), q_2^{(i+2)}(Y \mid \hat{Y})$$

and distributions $q_1^{(i+2)}(Y \mid \hat{x}), 1 \le \hat{x} \le r$ and $q_2^{(i+2)}(Y \mid \hat{z}), 1 \le \hat{z} \le t$ using $q_1^{(i)}(y \mid \hat{x}) = q_1^{(i)}(y \mid \hat{y}) q_1^{(i)}(\hat{y} \mid \hat{x})$, $q_2^{(i)}(y \mid \hat{z}) = q_2^{(i)}(y \mid \hat{y}) q_2^{(i)}(\hat{y} \mid \hat{z})$.

14. Stop and return $C_X = C_X^{(i+2)}$, $C_Y = C_Y^{(i+2)}$ and $C_Z = C_Z^{(i+2)}$ if the change in objective function value, that is,

$$\begin{aligned} &\alpha\left[D(p_1(X,Y)\,||\,q_1^{(i)}(X,Y)) - D(p_1(X,Y)\,||\,q_1^{(i+2)}(X,Y))\right] \\ &+ (1-\alpha)\left[D(p_2(Z,Y)\,||\,q_2^{(i)}(Z,Y)) - D(p_2(Z,Y)\,||\,q_2^{(i+2)}(Z,Y))\right], \end{aligned}$$

where $0 < \alpha < 1$ is small; else, set i=i+2 and go to step 2.

Overall speaking, at the beginning of each iteration of the CIT algorithm, we calculate the clusters of X and Y through the sub-problem of X-Y information theoretic co-clustering (see steps 2~6). Actually, instead of processing this sub-problem till it converges, we stop it after a few iterations and output the clustering of X. This is because that the aim of the above operation is to obtain a relatively good initial clustering of X for calculating the clustering of Y, rather than to get an accurate clustering of X. Similarly, the clusters of Z and Y are calculated through the sub-problem of Z-Y information theoretic co-clustering and the clustering of Z is outputted after several iterations (see steps 7~11). Then, the new clusters of Y are calculated under the concept of consistency, by minimizing the loss function (14) (defined in Definition 4). After that, the clusters of X, Y, and Z are all updated (Steps 12~14). This iterative process stops when the objective function no longer decreases.

Note that the convergence of this algorithm can be well proved. That is, this algorithm can monotonically decrease the objective function as given in Definition 4, and terminate in a finite number of iterations. We omitted the details of the proof due to the space restriction.

4. Experimental results

Web image clustering is a technology to help users digest the large amount of online visual information. Many traditional methods on image clustering only used either the low-level visual features inside the images or the surrounding text in the corresponding Web pages. Considering that these two kinds of information are complementary, one can expect better clustering results if we are able to utilize both of them in an effective way. Low-level visual features, images, and surrounding text can just make up a star structure where images are the central type of objects. Therefore, they can be well represented by the tripartite graph as shown in Figure 1.(b) and thus be solved by the CIT algorithm. In this section, we would like to show some evaluation results on this task.

4.1. Data preparation

The image data used in our experiments were crawled from the Photography Museums and Galleries of the Yahoo! Directory. Images and their surrounding texts were extracted from the crawled Web pages. After removing some low-quality data, the remaining 17,000 images were assigned to 48 categories manually.

In our experiment, we randomly selected 10 categories of images from this dataset, the names of which are listed in Table 3. We extracted 530-dimension color and texture features in total as the low-level visual representation of the images (See Table 2) to build the {visual feature}-by-image matrix A. To generate the {term in surrounding text}-by-image matrix B, we removed the stop words and then the remaining words were regarded as textual representations of the images in our experiments. The dimensionality of the textual features ranges from several hundreds to more than one thousand, changing with different subset of images.

Table 2. The low-level features extracted from images.

Feature category	Feature Name	Dimensions
Color	Color Histogram Features	256
	Color Moment Features	9
	Color Coherence Features	128
Texture	Tamura Texture Features	18
	Wavelet Features [2]	104
	MRSAR [9]	15

4.2. Average performance

We set parameter $\alpha = 0.4$ and report the clustering performance for all possible pairs of categories in the selected image dataset. We plot the comparison between CIT accuracy and CBGC accuracy in Figure 2, each point in which represents a category pair. We can see that most of the points fall in the upper side of the diagonal, indicating that the CIT algorithm outperforms the CBGC method in most cases. We also plot the comparison between CIT time cost and CBGC time cost in Figure 3, which shows definitely that the proposed algorithm is much more efficient than the CBGC algorithm. (In Figure 3, the indexes on the horizontal axis were a random permutation of all possible pairs of categories in the selected image dataset.) To sum up, the CIT algorithm is a better solution to the high-order co-clustering problem.

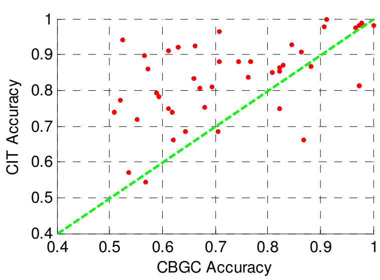

Figure 2. Accuracy comparison.

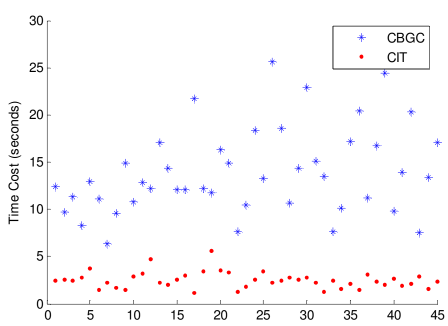

Figure 3. Time cost comparison.

5. Conclusions

In this paper, we proposed a novel algorithm based on the consistent information theory for co-clustering high-order heterogeneous data. This algorithm can be regarded as another realization of the consistency theory, and can also be regarded as an extension of the information-theoretic co-clustering algorithm. Experiments showed that it is a good choice for the co-clustering of multi-type inter-related data objects, in terms of both efficiency and effectiveness.

7. References

[1] Banerjee, A., Dhillon, I. S., Ghosh, J., Merugu, S., and Modha, D. S. A generalized maximum entropy approach to Bregman co-clustering and matrix approximations. In Proceedings of ACM SIGKDD'04, pages: 509-514, 2004.

[2] Chang, T., and Kuo, C.C.J. Texture analysis and classification with tree-structured wavelet transform. IEEE Transactions on Image Processing, vol. 2, no. 4, pages: 429-441, 1993.

[3] Cover, T. and Thomas, J. Elements of Information Theory. John Wiley & Sons, New York, USA, 1991.

[4] Dhillon, I.S. Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of ACM SIGKDD'01, pages: 269-274, 2001.

[5] Dhillon, I.S., Mallela, S., and Modha, D.S. Information-theoretic co-clustering. In Proceedings of ACM SIGKDD'03, pages: 89-98, 2003.

[6] El-Yaniv, R., and Souroujon, O. Iterative double clustering for unsupervised and semi-supervised learning. In Proceedings of ECML'01, pages: 121-132, 2001.

[7] Gao, B., Liu, T. Qin, T. Zheng, X. Cheng, Q. and Ma, W. Web image clustering by consistent utilization of visual features and surrounding texts. In Proceedings of ACM Multimedia'05, pages: 112-121, 2005.

[8] Gao, B., Liu, T., Zheng, X., Cheng, Q., and Ma, W. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In Proceedings of ACM SIGKDD'05, pages: 41-50, 2005.

[9] Mao, J. C., and Jain, A. K. Texture classification and segmentation using multiresolution simultaneous autoregressive models. Pattern Recognition, vol. 25, no. 2, pages: 173-188, 1992.

[10] Slonim, N., and Tishby, N. Document clustering using word clusters via the information bottleneck method. In Proceedings of ACM SIGIR'00, pages: 208-215, 2000.

[11] Wang, J., Zeng, H., Chen, Z., Lu, H., Tao, L., and Ma, W. ReCoM: reinforcement clustering of multi-type interrelated data objects. In Proceedings of ACM SIGIR'03, pages: 274-281, 2003.