Aggregate Two-Way Co-Clustering of Ads and User Data for Online Advertisements*

MENG-LUN WU, CHIA-HUI CHANG, RUI-ZHE LIU AND TENG-KAI FAN
Department of Computer Science and Information Engineering
National Central University
Taoyuan, 320 Taiwan

Clustering plays an important role in data mining, as it is used by many applications as a preprocessing step for data analysis. Traditional clustering focuses on grouping similar objects, while two-way co-clustering can group dyadic data (objects as well as their attributes) simultaneously. In this research, we apply two-way co-clustering to the analysis of online advertising where both ads and users need to be clustered. However, in addition to the ad-user link matrix that denotes the ads which a user has linked, we also have two additional matrices, which represent extra information about users and ads. In this paper, we proposed a 3-staged clustering method that makes use of the three data matrices to enhance clustering performance. In addition, an Iterative Cross Co-Clustering (ICCC) algorithm is also proposed for two-way co-clustering. The experiment is performed using the advertisement and user data from Morgenstern, a financial social website that focuses on the agency of advertisements. The result shows that iterative cross co-clustering provides better performance than traditional clustering and completes the task more efficiently.

Keywords: co-clustering, decision tree, KL divergence, dyadic data analysis, clustering evaluation

1. INTRODUCTION

Web advertising (Online advertising), a form of advertising that uses the World Wide Web to attract customers, has become one of the world’s most important marketing channels. Ad$Mart is an online advertising service launched by the Umach community website. Umach is a social networking platform that advertises “self-management” based on Web2.0 and is a content provider for financial planning. On one hand, Umach involves people through activities such as creative curriculum vitae, financial Olympia, world’s dynamic asset allocation, and other professional competitions; on the other hand, it runs the ad community platform “Ad$Mart” to share ad profits with users who place the ads on their personal homepages (called “Self-Portraits”) as advertising spokesmen. In contrast to the contextual advertising mechanism offered by Google Ad-sense, Umach services are built on a community entry with support for financial planning and analysis of users’ Lohas lifestyle for risk preferences.

Ad$Mart and Google Ad-sense both practice the long tail theory, which subverts the 80/20 law and provides low advertising costs. The biggest difference is that the users of Ad$Mart can choose what ads are placed on their self-portraits while the users of Google Ad-sense have hardly any control over what ads will be placed on their web sites. One

Received February 22, 2011; revised August 20, 2011; accepted August 31, 2011.
Communicated by Irwin King.

* This paper has been presented in the International Computer Symposium 2010 (ICS 2010) which was held in Tainan, Taiwan on Dec. 16-18, 2010 and sponsored by IEEE, NSC Taiwan, and MOE Taiwan.
benefit of this choice is that the users can maintain a consistent style for their homepages, which can avoid the inconsistent allocations of ads due to negative associations in given contexts, as shown in [9]. Although AdSMart has separated their business model from Google Ad-sense, the biggest issue here is how to provide advertisers with more specific flow and provide users with the most appropriate advertisements. In other words, the role of the ad agent is to create a triple win situation for advertisers, users and itself (i.e., the ad agent platform).

The main focus of this paper is to apply clustering techniques to group ads as well as users. The idea is to understand the correlations between ad clusters and user groups such that the agent platform could provide user group analysis for each ad cluster to determine the best strategy for them to maximize their advertising effectiveness. The traditional clustering problem deals with one-way clustering, which treats the rows of a data matrix as objects and uses the columns as attributes for similarity computation. Thus, if we want to obtain user groups and ad clusters, we may apply traditional clustering to the user-by-ad matrix to obtain user groups; then we may transpose the matrix to ad-by-user matrix and apply traditional one-way clustering to obtain ad clusters. Although such clustering results may be the best in terms of the inter-cluster over intra-cluster object distances, they do not consider the constraints that exist among ad clusters and user groups.

What makes the problem different is that we not only have the user-by-ad matrix but also two additional matrices, which describe specific features of users and ads. For example, the user-feature matrix of the Umatch community contains the result of lifestyle preference survey of the users, while the ad-feature matrix contains the settings of the advertisements (e.g., amount, and limitations on users). In this research, we design co-clustering algorithms for the analysis of online advertising where both ads and users need to be clustered.

The specific research questions to be addressed in this paper are:

1. How do we use the ad-feature matrix, the ad-by-user link matrix, and the user Lohas life mode analysis to achieve the co-clustering?
2. What kinds of measurements can verify the co-clustering results?

Following a discussion of the problem of co-clustering and data preprocessing, this paper reports on an exploratory study designed to investigate whether this kind of co-clustering algorithm can outperform traditional one-way clustering algorithms. The rest of the paper is organized as follows: section 2 introduces related work and some background knowledge; section 3 gives a more detailed description of the data used; section 4 describes the proposed algorithm; section 5 reports the experimental results; and finally, section 6 concludes the paper.

2. RELATED WORKS

This section briefly introduces the applications of one-way clustering in online advertising and the research on dyadic clustering.

One way clustering has been applied for both user and ad analysis. For user analysis, some prior studies [7] focused on how to construct a representative user profile (based on
the ads that he/she has clicked before) for user clustering. The ad agency can decide whether an incoming ad should be suggested to the users of specific group using a similarity measurement (e.g., the cosine function or the Jaccard function) between the ad and user group. Similarly, clustering algorithms have been applied to the ad-by-user matrix to discover ad clusters. Such ad clusters can then be used to determine whether a new ad should be assigned to highly active users within specific ad clusters.

Although there are well-developed one-way clustering algorithms, it often happens that the relationship between the row data and column data is useful. Therefore, two-way co-clustering has recently been proposed for grouping the dyadic data simultaneously. In many research studies, the idea of co-clustering is implemented by constraint optimization through some mathematical models. Dhillon et al. proposed “Information-theoretical co-clustering” [7], which regards the occurrence of words in documents as a joint probability distribution for the two random variables: the document and the word. By minimizing the difference in the mutual information (between document and word) before and after clustering and decomposing the objective function based on KL divergence, they are able to achieve the desired co-clustering through iterative assignment of documents and words to the best document and word clusters, respectively. Later, Banerjee et al. suggested a generalization maximum entropy co-clustering approach by appealing to Bregman information principle [1]. In addition to information theoretic based models, Long et al. (2005) utilized matrix decomposition to factorize the correlation matrix into three approximation matrices and achieve better co-clustering by optimization [2]. Meanwhile, Ding et al. gave a similar co-clustering approach based on nonnegative matrix factorization.


While the goal of this paper is co-clustering, we have two additional matrices, which are different from those used in the previous work. In the next section, we will give more detailed descriptions of the data.

3. DATA DESCRIPTION

AdSMart is an online ad service launched by Umatch as an activity of the financial social web-site. The goal of AdSMart is to create a triple-win commercial platform for the advertisers, registered users and platform provider. The idea is that an advertiser pays a low cost for its products to be shown on the website (e.g., members’ self-portraits), while the platform provider shares advertising profits with the registered members who link the ads on their web space to endorse the product, dividing the profits based on the activity scores of the registered members. The data used in this research are the advertisements that are displayed on AdSMart from September 2009 to March 2010.

There are three matrices available for our analysis: the ad-user link matrix, user Lohas matrix and ad-feature matrix. In contrast to traditional contextual advertising, users of AdSMart are asked to select ads before the beginning of displaying ads. Because the ads are sorted by the amount purchased (M) by advertisers when they are shown to
members for selection, the top ranked ads often have a lot of links by users. To avoid having all the profits go to a single individual with high popularity, the advertiser can set an upper bound on the users’ activity score (K_high). Similarly, the advertiser can set a lower bound on the activity score (K_low) to avoid the display of ads on unpopular self-portraits. In addition, advertisers can also limit the total number of users (L) linked to the ad. In summary, an ad may contain advertising settings, such as play date, title, daily amount (M), the order (O) in the ad list, the upper bound (K_high) and lower bound (K_low) of the users’ activity score, the number of users that can link the ad (L), and the resulting total number of linked people (N).

However, because each ad can have its own display schedule (from several days to several weeks), we also need some preprocessing steps to summarize various numbers of display dates or linked users. Therefore, we use six statistics: maximum, minimum, median, variance, 1st and 3rd quartile to summarize the various numbers of values into structured data. Meanwhile, we also apply the logarithm to these values for normalization. As for the bounds on users’ activity scores, we use a Boolean value to denote the case when the upper or lower bound is set (K = K_high ∪ K_low). Including the number of days that the ad is displayed, we have a total of 17 features for 530 ads.

In addition to the ad-feature matrix, we have two other data matrices for the analysis: the ad-by-user link matrix and the user Lohas matrix (user feature). The ad-by-user link data contains the number of ad links for a given user during 09/01/2009 to 03/31/2010, while the user Lohas matrix contains choices of users to 24 questions about their preferences to life styles. Although there are more than 30,000 members in the UMatch community, the total number of users who are also involved in Ad$Mart is only 9,860 and the total number of users who also played the Lohas game is only 21,883 during the relevant period. The intersection of the two sets is 2,124 users. Similarly, while more than six hundred ads have been displayed in Ad$Mart, only 530 ads are involved in this analysis.

4. THE PROPOSED CO-CLUSTERING METHODS

The goal of this paper is to analyze ad clusters as well as user groups. The objective is to determine if there is any connection between ad clusters and user groups. For example, are there special ad clusters that will be especially attractive to a particular group of users? Will some common settings of ads lead to large or small numbers of user links? Similarly, are there special user groups that will only link to particular clusters of ads? For example, will users who link to ads with high daily amounts also link to similar ads with high amounts?

Traditional clustering focuses on one-way clustering, which relies on the attributes of either the row or column, but not both. In this paper, we would like to achieve two-way clustering with two additional feature matrices. In a way, the two data matrices present new constraints on the 2-way clustering problem. This is why we could not apply existing co-clustering algorithms like information theoretic co-clustering [7], Bregman Co-clustering [1] or block value decomposition [2] since all these methods do not consider augmented matrices.

In this paper, we propose two clustering methods, three-stage clustering (3-stage)
and iterative cross co-clustering (ICCC). Both methods aggregate traditional one-way clustering, $K$-means, to overcome the high dimension problem when combined with an ad-by-user link (or user-by-ad link) matrix. The former aims to cluster either ads or users; thus, we can have either ad-based 3-stage clustering or user-based 3-stage clustering. In the following, we will introduce ad-based 3-stage clustering (section 4.1) and user-based three-stage clustering (section 4.2) separately. Finally, iterative cross co-clustering conducts two-way clustering for ads and users at the same time.

4.1 Ad-Based 3-Stage Clustering

As implied by the name of 3-stage clustering, there are three steps. For each step, we apply $K$-means on the prepared data matrix with a preset number ($= 1000$) of iterations. The first step use $K$-means on the ad-feature matrix to produce ad clusters that could be used to group user-ad links for user clustering in the second step. Finally, the user groups produced in the second step are used to group ad-user links for ad clustering in the third step. In other words, we make use of the ad-feature matrix first and then bind ad-feature matrix with the reduced link matrix in the clustering to produce ad clusters. Formally, we have the ad-feature matrix $A$ and ad-by-user matrix $L$ as our input.

**Step 1:** We adopt the ad-feature matrix as the input data of this step. The $K$-means algorithm is applied to the ad-feature matrix to generate the ad cluster set $\hat{A}$. The parameter $K$ is set to range from 2 to the logarithm of the data size.

**Step 2:** The ad clusters generated from step 1 are used to reduce the size of the ad dimension by summing up the link counts of user $u_j$ over all ads in ad cluster $\alpha_k$. Let $L_{ij}$ denotes the number of counts that ad $a_i$ has been linked by user $u_j$ during the period. We reduce the size of ad dimension by Eq. (1) to produce the user by ad-cluster matrix.

$$L_{U\times\hat{A}}(i, k) = \log_2 \left( \sum_{a_i \in \alpha_k} L_{ij} \right)$$

(1)

To prevent extremely large values from dominating the clustering results, we take the logarithm of the summation. We use the reduced matrix $L_{U\times\hat{A}}$ as the input data and apply the $K$-means algorithm to the input data. In this step, we obtain user group set $\hat{U}$, which will be the reference data of next step.

**Step 3:** In this step, we use the user groups $\hat{U}$ and ad-user link matrix to reduce the size of the user dimension for the ad-by-user matrix $L$: we sum up the click counts of ad $a_i$ over all users in user group $\nu_l$, as Eq. (2) shows:

$$L_{\hat{U}\times\hat{A}}(i, l) = \log_2 \left( \sum_{u_j \in \nu_l} L_{ij} \right)$$

(2)

After the reduction, we concatenate the reduced matrix $L_{\hat{U}\times\hat{A}}$ with the ad-feature matrix $A$. We then use the $K$-means algorithm to group this combined matrix and generate the new ad cluster $\hat{A}'$ as our final result. The complete algorithm of the ad-based 3-stage algorithm is shown in Fig. 1.
4.2 User-Based 3-Staged Clustering

Similarly, we can conduct user-based 3-stage clustering to obtain user groups by considering user Lohas matrix. In this part, the input data are the user Lohas matrix $U$ and the ad-by-user link matrix $L$. As mentioned above, the clustering of each step also uses the $K$-means algorithm.

**Step 1:** In this step, the ad-feature matrix is replaced with the user Lohas matrix $U$ to generate the user clustering result $\hat{U}$.

**Step 2:** Given the user groups $\hat{U}$ generated from step 1, we use Eq. (2) to reduce the size of the user dimension of the ad-by-user link matrix, producing the ad by user-group matrix $L_{A\times\hat{U}}$.

**Step 3:** Based on the ad clusters generated from step 2, we reduce the ad dimension of the user-by-ad link matrix $L^T_u$ to produce the user by ad-cluster link matrix $L_{U\times\hat{A}}$. Then, we concatenate the reduced matrix with the user Lohas matrix, which is used to group the users. The complete algorithm of the user-based 3-stage algorithm is shown in Fig. 2.

**Input:** User Lohas matrix ($U$) and ad-by-user link matrix ($L$).
1. Apply $K$-means to user matrix $U$ to get initial user group $\hat{U}$.
2. Ad clustering:
   (a) Merge the user dimension of the ad-by-user link matrix $L$ by user groups $\hat{U}$ using Eq. (2) to get $L_{U\times\hat{A}}$.
   (b) Apply $K$-means to $L_{U\times\hat{A}}$ matrix to get ad clusters $\hat{A}$.
3. User re-grouping:
   (a) Transpose $L$ to get user-by-ad matrix and merge the ad dimension by ad cluster $\hat{A}$ using Eq. (1) to get $L_{U\times\hat{A}}$.
   (b) Apply $K$-means to the combined $U + L_{U\times\hat{A}}$ matrix to get new user group $\hat{U}$.

Fig. 1. Ad-based 3-stage clustering algorithm.

Fig. 2. User based 3-stage clustering algorithm.
4.3 Iterative Cross Co-clustering

3-stage clustering only groups ads and users separately based on the ad-feature matrix or the user Lohas matrix with the ad-by-user link matrix, but it does not cluster both ad data and user data simultaneously. Iterative Cross Co-Clustering (ICCC) improves the 3-staged clustering results by taking both ad data and user Lohas matrix into account and cross-correlating the clustering results.

During the ICCC process, we also utilize the K-means algorithm to group the data in each step. We define the initial ad clustering and user grouping based on the grouping of the ad-feature matrix and user Lohas matrix separately. As in the 3-stage clustering process, we reduce the size of ad or user dimension by merging the link matrix based on Eqs. (1) and (2), respectively. The reduced matrix is then combined with the ad-feature matrix or the user Lohas matrix for clustering. We repeat this process until a stable result is generated. The complete algorithm is divided into four steps as described below.

**Step 1:** Given the ad-feature matrix and the user Lohas matrix, we first apply the K-means algorithm to obtain the initial ad clustering and user grouping, respectively.

**Step 2:** (Ad re-clustering) In this step, we merge the ad-by-user link matrix based on previous user grouping result $\hat{U}_t$. The reduction is conducted using Eq. (2) to obtain the ad by user-group link matrix $L_{A\times\hat{U}}$. We then apply K-means algorithm on the combined matrix $(A + L_{A\times\hat{U}})$ of the ad-feature data and the reduced link matrix for to get new ad clustering result $\hat{A}_{t+1}$.

**Step 3:** (User re-grouping) Similarly, we apply Eq. (1) to reduce the ad dimension of the user-by-ad link matrix based on previous ad clustering $\hat{A}_t$ and obtain the user by ad-group link matrix $L_{U\times\hat{A}}$. Then, we apply K-means algorithm on the combined matrix of the user Lohas data $U$ and the reduced matrix to generate a new user grouping $\hat{U}_{t+1}$.

**Step 4:** We use the new ad cluster $\hat{A}$ and user group $\hat{U}$ to repeat steps 2 and 3 until the clustering results reach a stable state. We use trial and error to get the best setting for the number of iterations. The complete algorithm is shown in Fig. 3 with illustrated flowchart in Fig. 4.

**Iterative Cross Co-clustering Algorithm**

**Input:** Ad feature matrix $(A)$, User Lohas matrix $(U)$, and ad-by-user link matrix $(L)$.

1. Initial clustering:
   (a) Apply K-means to ad matrix $A$ to get initial ad clusters $\hat{A}_0$.
   (b) Apply K-means to user matrix $U$ to get initial user groups $\hat{U}_0$.
   (c) $t := 0$.

2. Ad clustering:
   (a) Reduce the user dimension by Eq. (2) based on user groups $\hat{U}_t$ to get $L_{A\times\hat{U}}$.
   (b) Apply K-means to the combined matrix $A + L_{A\times\hat{U}}$ to get new ad clusters $\hat{A}_{t+1}$.

3. User grouping:
   (a) Reduce the ad dimension by Eq. (1) based on ad clusters $\hat{A}_t$ to get $L_{U\times\hat{A}}$.
   (b) Apply K-means to the combined matrix $U + L_{U\times\hat{A}}$ to get new user groups $\hat{U}_{t+1}$.

4. $t := t + 1$; Go to step 2.

Fig. 3. Iterative cross co-clustering algorithm.
5. EXPERIMENTAL RESULTS

In order to have fair comparison, we implement K-means, and Information-Theoretic Co-Clustering (ITCC) [7] as comparison methods. For K-means, we apply the algorithm twice for ad and user, respectively. First, we run K-means on the combined matrix of ad-feature and ad-by-user link matrix to obtain the base ad clusters $A_B$. Note that combined matrix has a total of 17 ad features plus 2,124 user features. Similarly, we also apply K-means on the combined matrix of user Lohas and user-by-ad link matrix to obtain the base user groups, where the combined matrix is a total of 24 Lohas plus 530 ad features.

We also implement the state-of-the-art co-clustering algorithm ITCC to group dyadic data simultaneously. We apply ITCC on the ad-by-user link matrix $L$ with initial ad cluster and user group generated by K-means on $L$ and $L^T$ separately. ITCC is an iterative process which assigns new ad cluster and user group for each ad and user to minimize the loss of their mutual information after co-clustering. Note that due to the algorithm property, ITCC can only operate on the link matrix, while other three algorithms are based on K-means of the combined matrix.

5.1 Evaluation Methods

Clustering methods are usually evaluated by classification accuracy based on some standard answers. The question is “when there are no standard answers, can we still use classification accuracy for evaluation metric?” In this paper, we propose two kinds of evaluation methods, the classification based evaluation and the KL divergence based evaluation. The former evaluates the performance of one-way clustering, while the later evaluates the performance of two-way co-clustering.

For one-way clustering, we use the average F-measure of the (cross-validation) classification results to indicate the performance of a clustering. We regard the clustering result as the classification target and apply Decision Trees to construct a model for classification. The idea is that the higher the average F-measure of the cross-validation, the better the clustering result. Thus, we can evaluate the performance of different clustering
methods even if we don’t have class labels for the objects. To avoid the bias of Decision trees, we also try several other classifiers including Simple logistic, \( k \)-NN, Naïve Bayes and SVM.

For two-way co-clustering, we exploit the characteristic of co-clustering by measuring the differences in the class distributions between clusters. Given \( k \) ad clusters \( \hat{A} = \{ \alpha_1, \alpha_2, ..., \alpha_k \} \) and \( l \) user groups, \( \hat{U} = \{ \upsilon_1, \upsilon_2, ..., \upsilon_l \} \), we can define the user group distribution \( p(\hat{U} | \alpha_i) \) for each ad cluster \( \alpha_i \), \( i \in [1, k] \) and compute the KL divergence between any two clusters \( \alpha_i \) and \( \alpha_j \) to represent the difference between the two clusters.

\[
D_{KL}(p(\hat{U} | \alpha_i) \| p(\hat{U} | \alpha_j)) = \sum_{n=1}^{l} p(\upsilon_n | \alpha_i) \cdot \log \left( \frac{p(\upsilon_n | \alpha_i)}{p(\upsilon_n | \alpha_j)} \right)
\]

Similarly, we can define the ad distribution \( p(\hat{A} | \upsilon_j) \) for each user group \( \upsilon_j, j \in [1, l] \) and compute the KL divergence between any two user groups \( \upsilon_i \) and \( \upsilon_j \) to represent the difference between these two user groups.

\[
D_{KL}(p(\hat{A} | \upsilon_i) \| p(\hat{A} | \upsilon_j)) = \sum_{m=1}^{k} p(\alpha_m | \upsilon_j) \cdot \log \left( \frac{p(\alpha_m | \upsilon_j)}{p(\alpha_m | \upsilon_i)} \right)
\]

The higher the KL divergence is, the better the co-clustering result we achieve. Thus, we could evaluate the co-clustering performance according to the average KL divergence of the ad clusters and user groups as follows.

\[
KL_{CC} = KL_{ad} + KL_{user}
\]

\[
KL_{ad} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j \neq i}^{K} D_{KL}(p(\hat{U} | \alpha_i) \| p(\hat{U} | \alpha_j))
\]

\[
KL_{user} = \frac{1}{L(L-1)} \sum_{i=1}^{L} \sum_{j \neq i}^{L} D_{KL}(p(\hat{A} | \upsilon_i) \| p(\hat{A} | \upsilon_j))
\]

5.2 Evaluation of Two-Way Clustering

To calculate the KL divergence of the co-clustering result, we reduce the link matrix \( L \) according to the result of ad cluster \( \hat{A} \) and user group \( \hat{U} \) to generate ad-cluster by user-group link matrix \( L_{\hat{A} \times \hat{U}} \) with the following equation.

\[
L_{\hat{A} \times \hat{U}}(k, l) = \sum_{\alpha_i \in \hat{A}} \sum_{\upsilon_j \in \hat{U}} L_{ij}
\]

When measuring KL divergence of two ad clusters \( \alpha_i \) and \( \alpha_j \), we take the \( i \)th and \( j \)th rows of \( L_{\hat{A} \times \hat{U}} \) and normalize these two vectors respectively to obtain \( p(\hat{U} | \alpha_i) \) and \( p(\hat{U} | \alpha_j) \). Similarly, when measuring the KL divergence of two user groups \( \upsilon_i \) and \( \upsilon_j \), we take the \( i \)th and \( j \)th columns of \( L_{\hat{A} \times \hat{U}} \) and normalize these two vectors respectively to obtain \( p(\hat{A} | \upsilon_i) \) and \( p(\hat{A} | \upsilon_j) \).

Fig. 5 shows the KL divergence of the four methods (given \( K = 5 \)). Although none
of these methods are designed to optimize $KL_{cc}$, 3-stage and ICCC present better KL divergence, while ITCC and the baseline approach present lower KL divergence. However, ICCC does not gain additional performance by extra iteration. Similar results can be obtained for various K as will be shown later in section 5.4.

5.3 Evaluation of One-Way Clustering

For evaluation of one-way clustering, we take the ad clustering result as the target labels for the training data (i.e. the combined matrix of ad-feature and the ad-by-user link matrix) and conduct 10-fold cross validation on 530 ads with 2,141 features (including 2,124 users and 17 ad-features).

The experimental result with $K = 5$ is shown in Table 1, where various classifiers including SVM, Simple logistic, Decision tree (J48), Naïve Bayess, and $k$-NN are used. As we can see, although the training process uses the same data matrix with the baseline approach, the F-measure of the 10-fold cross validation based on the baseline approach does not outperform that produced by 3-stage and ICCC approaches. In average, ICCC produces ad clusters, which have higher classification accuracy across various learning algorithm, a desirable feature for a clustering algorithm. Although ITCC has good reputation of formulation, they couldn’t gain the highest performance in all kinds of classifiers. On the other hand, even though the 3-stage clustering algorithm presents larger KL divergence in two-way clustering evaluation, it also has larger variance in terms of various classification algorithms, which is not desirable for a clustering algorithm.

Table 1. Performance of ad clustering with respect to various classifiers ($K = 5$).

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>Logistic</th>
<th>J48</th>
<th>Naïve</th>
<th>$k$-NN</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.933</td>
<td>0.962</td>
<td>0.915</td>
<td>0.437</td>
<td>0.743</td>
<td>0.798</td>
</tr>
<tr>
<td>ITCC</td>
<td>0.853</td>
<td>0.817</td>
<td>0.725</td>
<td><strong>0.810</strong></td>
<td><strong>0.793</strong></td>
<td>0.800</td>
</tr>
<tr>
<td>3-stage</td>
<td>0.958</td>
<td>0.955</td>
<td>0.932</td>
<td>0.617</td>
<td>0.764</td>
<td>0.845</td>
</tr>
<tr>
<td>ICCC</td>
<td><strong>0.992</strong></td>
<td><strong>0.977</strong></td>
<td><strong>0.947</strong></td>
<td>0.724</td>
<td>0.712</td>
<td><strong>0.870</strong></td>
</tr>
</tbody>
</table>
Similarly, we take the user grouping result as the target labels for the training data (i.e. the combined matrix of user Lohas and the user-by-ad link matrix) and conduct 10-fold cross validation on 2,124 users with 554 features (including 24 choices and 530 ad-features).

Table 2 illustrates the F-measure of the 10-fold cross validation conducted on the user data matrix given the labels produced by four clustering approaches mentioned above (for $K = 5$). The proposed method ICCC algorithm still shows better performance than the baseline in four of the five classification algorithms. Although user-based 3-staged algorithm doesn’t present better performance, the difference is not significant compare with the baseline approach. Note that ITCC performs poor in user grouping. Thus, we conclude that the clustering result of the ICCC is as classifiable as that produced by other four approaches.

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>Logistic</th>
<th>Naive</th>
<th>J48</th>
<th>$k$-NN</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.950</td>
<td>0.952</td>
<td>0.436</td>
<td>0.811</td>
<td>0.808</td>
<td>0.791</td>
</tr>
<tr>
<td>ITCC</td>
<td>0.210</td>
<td>0.204</td>
<td>0.191</td>
<td>0.226</td>
<td>0.219</td>
<td>0.210</td>
</tr>
<tr>
<td>3-stage</td>
<td>0.939</td>
<td>0.932</td>
<td>0.420</td>
<td>0.776</td>
<td>0.801</td>
<td>0.774</td>
</tr>
<tr>
<td>ICCC</td>
<td>0.971</td>
<td>0.955</td>
<td>0.840</td>
<td>0.815</td>
<td>0.804</td>
<td>0.877</td>
</tr>
</tbody>
</table>

5.4 Effects of Parameters

Next, we show the effect of various $K$ on one-way and two-way clustering evaluation, respectively. Figs. 6 and 7 show the average F-measure of ad classification and user classification for one-way clustering evaluation. As we can see, the ad clusters produced by ICCC is more classifiable for larger $K$ and are comparable with the clustering results of ITCC and baseline approach in smaller $K$ (Fig. 6). Although the user group produced by ITCC is more classifiable for $K = 2$, the F-measure degrades a lot for large $K$. As for F-measure of user classification, ICCC still shows better result than other three methods.

![Fig. 6. Average F-measures of ad classification with different $K$.](image-url)
Finally, for two-way clustering evaluation, the performances of the 3-stage algorithm and ICCC are often better than the baseline approach except for $K = 2$ as shown in Fig. 8 where the baseline approach has the largest $KL_{cc}$.

To summarize the above evaluation results, while the proposed approaches have better performance in two-way evaluation by $KL$-divergence, their clustering results are also as classifiable as the result produced by the baseline approach. Meanwhile, although the 3 stage approach can achieve higher $KL$-divergence, the ICCC algorithm shows much stable F-measure in one-way clustering evaluation.

6. CONCLUSION

Traditional clustering focuses on the one-way clustering methods that cannot fully
use the correlations among dyadic data. In this paper, we start from a simple 3-stage clustering which runs the $K$-means algorithm three times to make use of the ad-feature and the user Lohas matrix in addition to the ad-by-user link matrix. We also propose the Iterative Cross Co-Clustering algorithm, ICCC, to smooth the co-clustering result. Both algorithms make use of one-way clustering to achieve the co-clustering task by merging the link matrix based on clustering result to convey the co-clustering requirement into one-way clustering.

We propose two evaluation methods to compare the performance of different co-clustering algorithms. The $KL_c$ measures the differences between ad cluster distributions of user groups, as well as the difference between user group distributions of ad clusters; while the F-measure of 10-fold cross validation based on classification indicate the ease of building a classifier based on the clustering results. The experiment results show that the proposed 3-staged algorithm and ICCC are able to produce higher $KL_c$ while ICCC maintaining good F-measures. Although the proposed methods are somehow heuristic, the two-way clustering evaluation is better and one-way clustering is more stable than state-of-the-art co-clustering algorithm information-theoretic co-clustering algorithm ITCC.

REFERENCES

8. K. Sugiyama, K. Hatano, and M. Yoshikawa “Adaptive web search based on user profile constructed without any effort from users,” in Proceedings of the 13th Inter-

Meng-Lun Wu (巫孟倫) is a Ph.D. student in Computer Science and Information Engineering Department at National Central University in Taiwan. He received his M.S. in Computer Science and Information Engineering department from Aletheia University in 2009, Taiwan. His research interests include co-clustering, collaborative filtering, recommendation system and machine learning.

Chia-Hui Chang (張嘉惠) is an Associate Professor at National Central University in Taiwan. She received her B.S. in Computer Science and Information Engineering from National Taiwan University, Taiwan in 1993 and Ph.D. in the same department in January 1999. Her research interests include web information integration, knowledge discovery from databases, machine learning, and data mining.

Rui-Zhe Liu (劉睿哲) is a Master student in Computer Science and Information Engineering Department at National Central University in Taiwan. He received his B.S. in Computer Science and Information Engineering from National Central University in 2009, Taiwan. His research interests include co-clustering, collaborative filtering, recommendation system and machine learning.
Teng-Kai Fan (范登凱) received the Ph.D. degree in Institute of Computer Science and Information Engineering Department at National Central University in 2010, Taiwan. His research interests include information extraction, text analysis, knowledge discovery from databases, and machine learning.