Cognitive Computation (2021) 13:504–517
https://doi.org/10.1007/s12559-021-09827-8

Co-Adjustment Learning for Co-Clustering

Ji Zhang[1] · Hongjun Wang[1] · Shudong Huang[1] · Tianrun Li[1] · Peng Jin[1] · Ping Deng[1] · Qigang Zhao[1]

Received: 8 August 2019 / Accepted: 12 January 2021 / Published online: 18 January 2021

Abstract
Co-clustering simultaneously performs clustering on the sample and feature dimensions of the data matrix, so it can obtain better insight into the data than traditional clustering. Adjustment learning extracts valuable information from chunklets for unsupervised cluster learning in specific scenarios, but in fact it can be easily extended to semi-supervised and supervised learning situations. In this paper, we propose a novel co-clustering framework, named co-adjustment learning for co-clustering (CALCC), and CALCC can be simultaneously used in unsupervised, semi-supervised and supervised learning situations. A novel co-adjustment learning (CAL) model is proposed to extract meaningful representations in both sample space and feature space for co-clustering. CAL can not only perform the sample projection as well as feature projection under the guidance of chunklet information, it can also transform the original data into another space with improved separability. We can obtain the row partition matrix and column partition matrix by performing the clustering process on the representations learned by the CAL model. In order to prove the availability of our framework, an unsupervised case of CALCC is introduced to make an extensive comparison with several related methods (specifically including the classic co-clustering methods and the state-of-the-art methods closely related to our work) on several image and real data sets. The experimental results show the superior performance of the CAL model in discovering discriminative representations and demonstrate the effectiveness of the CALCC framework. The proposed CALCC framework, as demonstrated in the experiments, is more effective superior to the related methods. In addition, the chunklet information can be effective to enhance the expression ability of the learned representations.

Keywords Co-clustering · Chunklet constraints · Co-projection · Representation learning

Introduction

Co-clustering (or bi-clustering [10]) is a widely used and powerful unsupervised learning solution that simultaneously performs clustering on rows and columns of a data matrix to explore inter-correlated patterns. Unlike traditional clustering methods [1] that aim to group rows or columns of the data matrix into clusters, co-clustering is intended to reorganize the original data matrix into blocks (i.e., co-clusters). Specifically, co-clustering describes the partitioning of the original data matrix into k row-clusters and l column-clusters (i.e., the total number of co-clusters is $k \times l$) using similarity measures according to a certain evaluation criterion. According to the criterion, the similarity between two instances from the same cluster is higher than that of instances from different clusters [21]. This approach differs from subspace clustering, which focuses on selecting a quantity of original dimensions in some unsupervised manner such that cluster structures become more evident in

✉ Hongjun Wang
wanghongjun@swjtu.edu.cn

Ji Zhang
jizhang@my.swjtu.edu.cn

Shudong Huang
huangshuddong@my.swjtu.edu.cn

Tianrun Li
trli@swjtu.edu.cn

Peng Jin
jandp@pku.edu.cn

Ping Deng
dengping609@gmail.com

Qigang Zhao
qgzhao@swjtu.edu.cn

[1] School of Information Science and Technology, Southwest Jiaotong University, 611756 Chengdu, P.R. China

this subspace [12]. Since co-clustering algorithms utilize the relations between sample clusters and feature clusters, they make the data sets more predictable and the co-clustering performance more excellent compared with traditional one-side clustering.

However, it is always a challenging task to achieve high-performance co-clustering quality without background information. To consider the prior information, semi-supervised co-clustering was proposed [34]. Current semi-supervised co-clustering methods focus on incorporating the known prior knowledge into the co-clustering algorithms so that the co-clustering performance can be improved. Computationally, most semi-supervised co-clustering algorithms lack flexibility because the constraints must be satisfied at each stage. Moreover, those semi-supervised co-clustering methods only consider the given constraints to be satisfied and clustering in the original data space. They are incapable of transforming the original data into a lower-dimensional data space guided by constraints. Constraint projection is a prevalent technique to address this problem in semi-supervised clustering [13].

Constraint projection (CP) aims to transform the original data into a lower-dimensional data space guided by constraints (usually pairwise constraints). The reduced data can still remain as the original class information. Recently, CP technology has also been integrated with semi-supervised co-clustering algorithms. Constraint co-projection can transform the original sample space and feature space into a low-dimensional space through simultaneously performing the sample CP (SCP) as well as feature CP (FCP) [19]. CP has been successful in incorporating prior knowledge into the representation learning process of the original data. However, a major challenge facing the CP is its high dependence on prior knowledge. In many tasks, even a small amount of prior knowledge is difficult to obtain owing to the high cost of the data-labeling process. It is desirable to apply the technology of CP to situations where prior knowledge cannot be directly accessed.

Adjustment learning is a simple and efficient approach that uses chunklets for unsupervised cluster learning in specific scenarios [33]. Chunklets are small groups of points that come from the unknown but the same class. Unlike labels, chunklets can sometimes be automatically obtained without human intervention. For each chunklet, the class label is consistent but unknown for all the data points belonging to it. This means that each class of data points consists of one or more chunklets in adjustment learning. Since chunklet information is not extracted from prior labeled data, adjustment learning is cast into the domain of unsupervised learning. However, adjustment learning can be easily extended to semi-supervised and supervised learning situations. In the first case, chunklets directly come from background information, i.e., we know in advance that a small number of samples or features (one or more) belong to the same class. In the second case, chunklets are extracted from the fully labeled data. Therefore, chunklet constraints can be regarded as another form of positive constraints. Unlike the must-link constraints, which hold that two instances must be clustered in the same cluster, the chunklet constraints specify instances belonging to the same cluster that do not necessarily appear in pairs; rather, they appear in groups.

In this paper, a novel co-clustering framework, named co-adjustment learning for co-clustering (CALCC), is proposed. Development of the framework was inspired by the following two aspects: i) Chunklets as a new form of positive constraint information can be effectively incorporated into the representation learning process of the CP model. ii) Simultaneous use of sample chunklets and feature chunklets can be helpful to obtain better insight into the data. In order to transform the original data space into another space with the property of better separability such that the clustering performance can be effectively enhanced, a novel co-projection model, named co-adjustment learning (CAL), is proposed. The sample chunklet constraints and feature chunklet constraints can be simultaneously used to guide the representation learning process in the proposed CAL model. After CAL is defined, we find that it is naturally born for co-clustering, so the CALCC co-clustering framework is proposed.

The proposed CALCC framework can be simultaneously used in unsupervised, semi-supervised and supervised learning situations. i) Unsupervised CALCC is designed to handle a frequently encountered problem. That is, in many tasks, it is difficult to find the clear range of a cluster, whereas it is easy to find the clear range of a chunklet, regardless of whether it resides in sample space or feature space. Thus, we can use different unsupervised clustering algorithms to generate clearly structured chunklets on both dimensions of the data matrix according to different characteristics of the data. ii) Semi-supervised CALCC can effectively address the situation: We know in advance that some samples or features belong to the same class (the groups of instances belonging to the same class can be seen as chunklets). iii) As mentioned above, chunklets are small groups of instances from the unknown but the same class. In supervised CALCC, we divide the fully labeled samples and features into sample chunklets and feature chunklets, respectively. The entire framework of CALCC is illustrated in Fig. 1.

To the best of our knowledge, this is the first work to improve the expression ability of the data space by incorporating the chunklet information into the representation learning process of the CP model. Several aspects of the paper are highlighted as follows:

i) We focus on utilizing the valuable chunklet information to simultaneously enhance the separability of the sample space and feature space of the original data. And the valuable chunklet information can be generated in unsupervised, semi-supervised and supervised settings.
ii) The proposed CAL model can simultaneously perform the sample projection as well as feature projection under the guidance of those chunklet information. In the transformed new representation space, the relations between samples and features are properly preserved and the input data become more predictable.
iii) Because unsupervised CALCC is the most difficult and valuable case in our framework, the representative CALCC-KM algorithm was proposed for unsupervised clustering task. In addition, we performed a comparative experiment on the three modalities of unsupervised CALCC. The results on benchmark data sets show the superior performance of the proposed framework.

The remainder of this paper is organized as follows: In Section 2, we introduce existing works on which our approach is based. In Section 3, we provide a detailed illustration of the proposed CAL model. Experimental results are shown in Section 4. The paper is concluded in Section 5.

Related Work

In this section, we review previous research closely related to our work. We first basically overview representative co-clustering algorithms. Then, we briefly introduce the technology of constraint projection.

Co-clustering has received significant attention from researchers since it was first proposed in the early 1970s [16]. Co-clustering is the most widely employed clustering approach in the fields of gene expression [7, 10, 14], natural language processing [8, 38] and recommender systems [17, 24].

Dhillon et al. [12] proposed a spectral co-clustering method (SCC) by which the document collection is modeled as a document-word bipartite graph. Accordingly, the co-clustering problem is regarded as a bipartite-graph partitioning problem. The background information is considered so that the clustering performance can be improved. Shi et al. [34] proposed a novel semi-supervised spectral co-clustering method (SCM). SCM can efficiently solve the poor clustering performance problem of most co-clustering algorithms caused by the sparsity of data and presence of noise. Numerous co-clustering methods are founded on information-theory-based models. The information theoretic co-clustering algorithm (ITCC) [11] seeks to enhance interrelated mutual information by performing simultaneous clustering on both column and row dimensions at each stage. Banerjee et al. [2] proposed a more general co-clustering framework wherein any Bregman divergence can be used in the objective function. Soon thereafter, Bekkerman et al. [3] extended the co-clustering framework to multi-way clustering to cluster a set of objects by simultaneously clustering their heterogeneous components. Moreover, Bayesian co-clustering (BCC) [31] and nonparametric Bayesian co-clustering ensembles [35] enable a mixed membership in column-clusters and row-clusters. Non-negative matrix factorization (NMF) and its graph-regularized extensions have received tremendous research interest over the past several years [18]. Chen et al. [9] presented a semi-supervised NMF method for

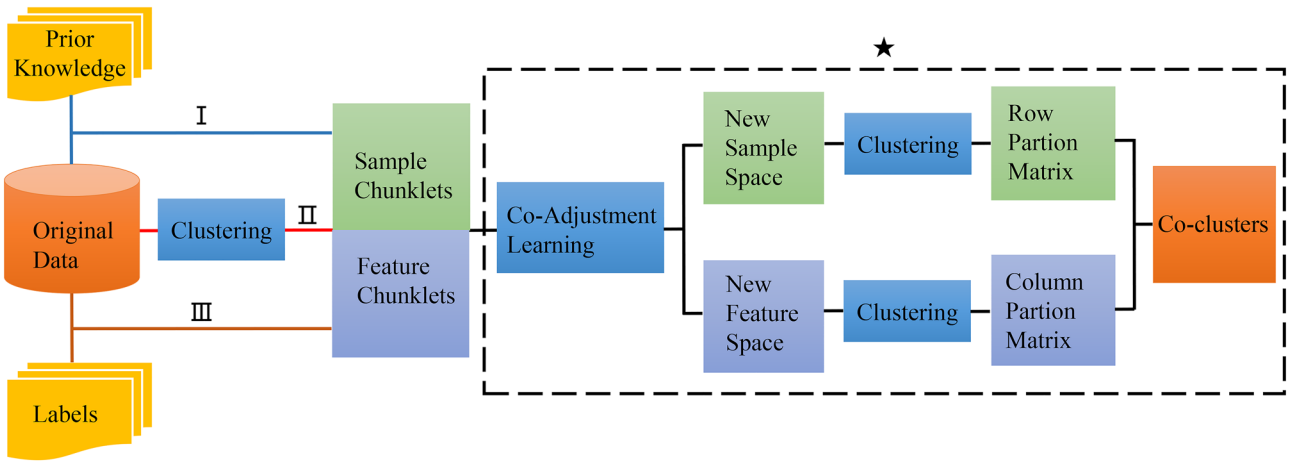

Fig. 1 Framework of CALCC. $I + \star$ = semi-supervised CALCC, $II + \star$ = unsupervised CALCC and $II + \star$ = supervised CALCC. Colored connecting lines in different colors are used to represent different ways to obtain chunklets

co-clustering (SSNMFCC). In this scheme, relational matrices can be computed through simultaneous modality selection and distance metric learning. More recently, Kumar et al. [23] presented a model-based co-clustering transfer learning algorithm to address data shifting problems. Whang et al. [37] presented non-exhaustive, overlapping co-clustering (NEO-CC), an effective solution to non-exhaustive, overlapping problems in co-clustering. Huang et al. [20] proposed a document co-clustering framework with adaptive local structure learning (ALSLCC) in which tri-factorization and intrinsic structure learning can be simultaneously performed. Nie et al. [27] proposed a novel co-clustering algorithm (SOBG) to learn a structured optimal bipartite graph with exactly k connected components from the original data, where k equals the number of clusters.

A large majority of CP methods aim to transform the original instances into a new low-dimensional space guided by the given constraints such that the clustering or classification performance can be improved in the new representation space. Zhang et al. [40] applied CP to ensemble learning. They first transformed the original data points into a new data space by using CP. Then, the base classifiers are built in this new space. Combined with pairwise constraints, a two-side CP method called constraint co-projection was furthermore proposed [19]. Constraint co-projection simultaneously performs SCP and FCP to project the original sample space and feature space into low-dimensional space. Moreover, relying on fully labeled data, a supervised subspace projections method was presented for constructing ensembles of classifiers [15]. The algorithms based on using CP for classification are described in [29] and [28].

Despite the widespread application of CP in semi-supervised clustering, research on unsupervised CP remains limited and preliminary. Some previous research focused on using equivalence constraints as side information [22, 32]. Chunklets are a kind of equivalence constraint that can be automatically obtained without human intervention in many specific tasks. Adjustment learning is an approach that uses chunklets for unsupervised learning. However, only in specific scenarios in which chunklets can be naturally generated can adjustment learning be classified as an unsupervised setting. It is desirable to extend it to general scenarios to enable chunklet constraints to be used for CP. Furthermore, it is easy to find chunklets with an evident structure in both a sample space and feature space using appropriate unsupervised clustering methods. The proposed framework can flexibly employ different unsupervised clustering algorithms according to different data characteristics. Accordingly, it can generate sample chunklets and feature chunklets on the sample and feature dimensions of the data matrix, respectively. Based on the chunklets we obtain, original samples and features can be simultaneously projected into low-dimensional space using chunklet co-projection. Hence, in the new low-dimensional representation space, instances from the same chunklet are close to each other.

Co-Adjustment Learning Model

In this section, the proposed method is described in detail. Specifically, we formulate the objective function of the CAL model and find the solution to it. The algorithm of unsupervised CALCC is also detailed at the end of this section.

Problem Formulation

As defined above, co-clustering seeks coherent blocks of rows and columns to explore inter-correlated patterns of the data. To devise a good co-clustering framework, one must first characterize the "goodness" of a co-clustering framework. Let $X = (x_{ij})_{n\times p}$ be a data matrix of n rows and p columns in some input space. Let the two collections, $R = (R_1, R_2, ..., R_k)$ and $C = (C_1, C_2, ..., C_l)$, respectively, denote the partition of the set of rows and the partition of the set of columns of the data matrix, where k is the row-cluster number and l is the column-cluster number. At this point, the pair $(R_\iota, C_\jmath)$ is called the co-cluster $(\iota = 1, 2, ..., k,\ \jmath = 1, 2, ..., l)$. Let $N_{cc} = k \times l$ be the total number of co-clusters, $|.|$ denote the cardinality of a set. The quality of the co-clustering method is assessed by the total variance of co-clusters, which is denoted as $T(R_\iota, C_\jmath)$. Here,

$$T(R_\iota, C_\jmath) = \sum_{\tau=1}^{N_{cc}} \sum_{i\in R_\iota} \sum_{j\in C_\jmath} (x_{ij} - \mu_\tau)^2 \tag{1}$$

where μ_τ is the average value in the τ-th co-cluster and

$$\mu_\tau = \frac{\sum_{i\in R_\iota} \sum_{j\in C_\jmath} x_{ij}}{|R_\iota||C_\jmath|}.$$

Generally speaking, a co-clustering framework having a zero total variance is considered perfect, and a co-clustering framework having a lower total variance is better than that with a higher total variance [5].

To obtain a lower total variance for our framework, we strive to simultaneously project the original samples and features into low-dimensional representation space guided by the chunklet constraints. Accordingly, the instances from the same chunklet are close to each other in the new low-dimensional space. Let the collection $C = \{x_{11}, ..., x_{1m_1}, \ldots, x_{c1}, ..., x_{cm_c}, \ldots, x_{\Omega 1}, ..., x_{\Omega m_\Omega}\}$ denote

the set of data points consisting of Ω chunklets, where $c = 1, ..., \Omega$ and x_{ci} denotes the i-th point in the c-th chunklet and m_c is the number of data points in the c-th chunklet. Let $m = \sum_{c=1}^{\Omega} m_c$. The within-chunklet covariance matrix W is defined as

$$W = \sum_{c=1}^{\Omega} \sum_{i=1}^{m_c} (x_{ci} - x_c)(x_{ci} - x_c)^T \tag{2}$$

where x_c is the mean of the c-th chunklet, θ^T denotes the transposition of vector θ.

The basic idea of the CAL model is to simultaneously transform the original samples and features into low-dimensional space guided by constraint information extracted from chunklets. Let $C_s = \{(s_i, s_j) | s_i$ and s_j be samples belonging to the same chunklet$\}$, $C_f = \{(f_i, f_j) | f_i$ and f_j are features belonging to the same chunklet$\}$. We define the sample within-chunklet covariance matrix W_s as follows:

$$\begin{aligned} W_s &= \frac{1}{|C_s|} \sum_{(s_i,s_j)\in C_s} [(s_i - \frac{s_i + s_j}{2})(s_i - \frac{s_i + s_j}{2})^T \\ &\quad + (s_j - \frac{s_i + s_j}{2})(s_j - \frac{s_i + s_j}{2})^T] \\ &= \frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} (s_i - s_j)(s_i - s_j)^T. \end{aligned} \tag{3}$$

Similarly, the feature within-chunklet covariance matrix W_f is defined as:

$$W_f = \frac{1}{2|C_f|} \sum_{(f_i,f_j)\in C_f} (f_i - f_j)(f_i - f_j)^T. \tag{4}$$

The objective of the CAL model is to learn two sets of co-projective matrix $U = [u_1, u_2, ..., u_L] \in R^{p\times L}$ and $M = [m_1, m_2, ..., m_G] \in R^{n\times G}$, which can simultaneously map the original samples and features into low-dimensional representation space. Hence, the original class information can be most reliably retained in the reduced data. At the same time, instances in the same sample (or feature) chunklet are close in the new low-dimensional sample (or feature) space. Thus, the CAL optimization problem is defined as minimizing the objective function $J(U, M)$. Moreover,

$$\begin{aligned} J(U, M) = &\frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} \| U^T(s_i - s_j) \|^2 \\ &+ \frac{1}{2|C_f|} \sum_{(f_i,f_j)\in C_f} \| M^T(f_i - f_j) \|^2 . \end{aligned} \tag{5}$$

The objective seeks to explore the inter-correlated patterns of samples and features. Based on it, we can learn two sets of co-projective matrix U and M, which can effectively project the original samples and features into low-dimensional sample space and feature space, respectively. We found that the new data space learned by the CAL model can not only capture data distributions, but also render the entities of the same chunklet become closer, while the entities of the different chunklets become farther apart.

Inference

To enhance the convenience of the subsequent discussions, we define S_s,S_f as follows:

$$S_s = \frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} \| U^T(s_i - s_j) \|^2 \tag{6}$$

and

$$S_f = \frac{1}{2|C_f|} \sum_{(f_i,f_j)\in C_f} \| M^T(f_i - f_j) \|^2 . \tag{7}$$

From objective function (5), if we intend to minimize $J(U, M)$, both S_s and S_f should be minimized. Furthermore, an analytical solution can be obtained for simultaneously finding the optimal co-projective matrix U in (6) and M in (7). First, we rewrite (6) as:

$$\begin{aligned} S_s =& \frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} \| U^T(s_i - s_j) \|^2 \\ =& \frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} \sum_{l}^{L} U_l^T(s_i - s_j)(s_i - s_j)^T U_l \\ =& \sum_{l}^{L} U_l^T \left(\frac{1}{2|C_s|} \sum_{(s_i,s_j)\in C_s} (s_i - s_j)(s_i - s_j)^T \right) U_l \\ =& \sum_{l}^{L} U_l^T W_s U_l. \end{aligned} \tag{8}$$

According to (8), S_f is rewritten as:

$$\begin{aligned} S_f =& \frac{1}{2|C_f|} \sum_{(f_i,f_j)\in C_f} \| M^T(f_i - f_j) \|^2 \\ =& \sum_{g}^{G} M_g^T W_f M_g. \end{aligned} \tag{9}$$

In this paper, we call S_s and S_f the chunklet scatter values. The most important step for the proposed CAL model is to seek two sets of co-projective matrix $U = [u_1, u_2, ..., u_L] \in R^{p\times L}$ and $M = [m_1, m_2, ..., m_G] \in R^{n\times G}$, so that the information in the sample chunklets and feature chunklets can be most effectively retained in the low-dimensional representation space. From (8) and (9), the objective function (5) is transformed into

$$\begin{aligned} J(U,M) =& \frac{1}{2|\mathcal{C}_s|}\sum_{(s_i,s_j)\in\mathcal{C}_s} \| U^T(s_i - s_j) \|^2 \\ &+ \frac{1}{2|\mathcal{C}_f|}\sum_{(f_i,f_j)\in\mathcal{C}_f} \| M^T(f_i - f_j) \|^2 \\ =& \sum_{l}^{L} U_l^T W_s U_l + \sum_{g}^{G} M_g^T W_f M_g. \end{aligned} \tag{10}$$

To simplify the problem, we assume $L = G$. Nevertheless, by a simple parameter setting, the proposed framework can easily solve the special case of $L \neq G$, which is discussed in Section 3.4.

According to simple algebraic theory [40], we further rewrite the objective function in (10) as:

$$\begin{aligned} J(U,M) =& \sum_{l}^{L} U_l^T W_s U_l + \sum_{g}^{G} M_g^T W_f M_g \\ =& Trace\left(U^T W_s U\right) + Trace\left(M^T W_f M\right) \\ =& Trace\left(\begin{bmatrix} U \\ M \end{bmatrix}^T \begin{bmatrix} W_s & 0 \\ 0 & W_f \end{bmatrix} \begin{bmatrix} U \\ M \end{bmatrix}\right) \\ =& Trace\left(V^T W V\right) \end{aligned} \tag{11}$$

$$s.t \quad V^T V = I$$

where $V = [U;M]$, and

$$W = \begin{bmatrix} W_s & 0 \\ 0 & W_f \end{bmatrix}.$$

To find the solution to this optimization problem, the traditional Lagrange multiplier optimization technique [19] is used. The Lagrangian can be denoted as:

$$L_{V_1,\ldots,V_\eta} = \hat{J}\left(V_1,\ldots,V_\eta\right) - \sum_{\varepsilon=1}^{\eta} \delta_\varepsilon \left(V_\varepsilon^T V_\varepsilon - 1\right). \tag{12}$$

We calculate the partial derivative of $L_{V_1,\ldots,V_\eta}$ with respect to each V_ε and set it to zero. Hence,

$$\begin{aligned} &\frac{\partial L}{\partial V_\varepsilon} = 2WV_\varepsilon - 2\delta_\varepsilon V_\varepsilon = 0, \quad \forall_\varepsilon = 1,\ldots,\eta \\ &\Rightarrow WV_\varepsilon = \delta_\varepsilon V_\varepsilon, \quad \forall_\varepsilon = 1,\ldots,\eta. \end{aligned} \tag{13}$$

It is apparent from (13) that the solution V_ε is an eigenvector of W, and δ_ε is the corresponding eigenvalue of W. Therefore, to minimize J, V must be composed of the first λ eigenvectors of W, which makes J the sum of the λ smallest eigenvalues of W.

Suppose $\Lambda_d = [\gamma_1 \leq,\ldots,\leq \gamma_d]$ is the solution to (13), and $V_d = [V_1,\ldots,V_d]$ is the corresponding eigenvector matrix. Denote the within-chunklet covariance matrix $W' = diag(\gamma_1,\gamma_2,\ldots,\gamma_d)$. Thus, the optimization problem is transformed into a trace minimization problem. Its solution is to choose i for:

$$Minimize\left(\sum_i \gamma_i\right). \tag{14}$$

Equation (14) indicates that it can be minimized when we choose the non-negative eigenvalues of γ_i for the sum. That is, we take the first d smallest non-negative eigenvalues of W, and we minimize $J(U, M)$ by constructing $V_d = [U;M]$ using the corresponding eigenvectors. Additionally, we have:

$$W' = V_d^T W V_d = \Lambda_d. \tag{15}$$

Now, we obtain the co-projection matrices U and M, and the solution to the optimization is found.

However, to obtain the optimal rescaling transformation [36], it is desirable to let the within-chunklet covariance matrix be fixed, i.e., we let it be the identify matrix. To that end, equation (15) is reconstructed as:

$$\begin{aligned} W' &= V_d^T W V_d \\ &= \Lambda_d^{-\frac{1}{2}} (V_d^T W V_d) \Lambda_d^{-\frac{1}{2}} \\ &= (V_d \Lambda_d^{-\frac{1}{2}})^T W (V_d \Lambda_d^{-\frac{1}{2}}) = I. \end{aligned} \tag{16}$$

Thus, the final co-projective matrix of CALCC is defined as F^T, and

$$F = V_d \Lambda_d^{-\frac{1}{2}}. \tag{17}$$

Based on F, we can obtain the final co-projection matrices, U and M. Specifically, $U = F[F_1;F_2;\ldots;F_p]$ (F_i denotes the i-th row of F), which is the sample projection matrix with p rows and d columns. Meanwhile, $M = F[F_{p+1};F_{p+2};\ldots;F_{p+n}]$ is the feature projection matrix with n rows and d columns. After obtaining the final co-projective matrix $U = [u_1, u_2, \ldots, u_d] \in R^{p\times d}$ and $M = [m_1, m_2, \ldots, m_d] \in R^{n\times d}$, and by performing the sample-projection $Y_s = XU$ and feature-projection $Y_f = X^T M$, we can transform the original sample space and feature space into a low-dimensional sample space and feature space. Instances from the same chunklet become closer to each other in the learned representation space. In the new sample space and feature space, we can obtain the row partition matrix and column partition matrix from which coherent co-clusters can be found by running the unsupervised clustering algorithm. Consequently, we obtain the final co-clusters with the lowest total variance.

CALCC Co-Clustering Framework

Under the guidance of chunklet information, the proposed CAL model can transform the original sample

space and feature space into a new meaningful sample space and feature space, respectively. So it is naturally born for co-clustering framework. Hence, a novel co-clustering framework, named co-adjustment learning for co-clustering (CALCC), is proposed.

The proposed framework can be used in unsupervised, semi-supervised and supervised learning situations. In unsupervised CALCC, we flexibly choose the befitting unsupervised clustering algorithm to generate sample chunklets and feature chunklets according to the characteristics of the input data. In semi-supervised CALCC, chunklets directly come from prior knowledge. Chunklets can be extracted from the fully labeled samples and features in supervised CALCC.

While the performance of supervised CALCC is notable, reliance on such a large number of labeled data items limits the utility of CALCC in many domains where such data sets are not available. On the other hand, in many situations, there is no a priori knowledge that can be exploited, which makes semi-supervised learning useless. Thus, we herein focus on unsupervised CALCC.

Algorithm Description

According to the previously given inference, the detailed algorithm procedure for unsupervised CALCC is summarized in Algorithm 1. It should be noted that we assume $L = G$ for simplicity in this paper. In many cases, however, the dimensions of samples and features tend to be different for different data sets. Here, by a simple parameter setting in step 5, the corresponding dimensions of U and M can be differently set in accordance with specific needs.

Algorithm 1: Learning algorithm for unsupervised CALCC framework

Input: Data matrix $X_{n\times p}$; row cluster number k, row chunklet number $\tilde{k}$, column cluster number l, column chunklet number $\tilde{l}$.

Output: The partition matrix of row R, the partition matrix of column C.

1: Choose an appropriate unsupervised clustering algorithm to generate $\tilde{k}$ row chunklets and $\tilde{l}$ column chunklets from data matrix $X_{n\times p}$;
2: Compute W according to (3), (4) and (11);
3: Compute the eigenvalues and eigenvectors of W;
4: Construct F by the selected eigenvalues and corresponding eigenvectors according to (12), (13), (14) and (17);
5: Construct $U = F[F_1; F_2; ...; F_p]$ and $M = F[F_{p+1}; F_{p+2}; ...; F_{p+n}]$;
6: Compute $Y_s = X \times U$ and $Y_f = X^T \times M$;
7: Obtain the row partition matrix R and column partition matrix C by running the chosen clustering algorithm on Y_s and Y_f, respectively.

Experiments

This section evaluates unsupervised CALCC using several real-world data sets. In detail, CALCC-KM, a representative unsupervised case of CALCC, which uses k-means clustering for chunklet generation, is first introduced to make a performance comparison with several related co-clustering methods. Then, we evaluate the effectiveness of three modalities in unsupervised CALCC.

Datasets

We describe our experiments performed on several real-world data sets, including 10 image data sets from Microsoft Research Asia Multimedia (MSRA-MM) [25] and 9 real data sets from the University of California, Irvine (UCI), machine learning repository [4]. The summary of these 19 data sets is given in Table 1. The first 12 data sets (including 10 image data sets and 2 big real data sets) in the table were used to conduct a performance comparison of the representative CALCC-KM and several related co-clustering methods, and the last 7 data sets were used to demonstrate the effectiveness of unsupervised CALCC in various modalities.

To evaluate the availability of the proposed framework, we perform an extensive comparison of the effective CALCC-KM with several related methods which stay either classic or state of the art, specifically including

Table 1 Summary of data sets

No.	Data set	Source	Samples	Features	Classes
1	MM01	Microsoft	930	428	2
2	MM02	Microsoft	880	428	3
3	MM03	Microsoft	1100	428	2
4	MM04	Microsoft	840	428	4
5	MM05	Microsoft	968	428	2
6	MM06	Microsoft	1730	428	3
7	MM07	Microsoft	883	428	3
8	MM08	Microsoft	891	899	3
9	MM09	Microsoft	880	892	3
10	MM10	Microsoft	892	892	3
11	Noma	UCI	34465	118	2
12	Huma	UCI	10299	501	6
13	Ionosphere	UCI	354	34	2
14	Sona	UCI	208	60	2
15	Spect	UCI	267	22	2
16	Syncon	UCI	600	60	10
17	Vote	UCI	435	16	2
18	Wdbc	UCI	563	30	2
19	Credi	UCI	690	15	2

- SCC [12]: a spectral co-clustering method, in which the document collection is modeled as a document-word bipartite graph.
- ITCC [11]: an information theoretic co-clustering method aiming to enhance inter-related mutual information.
- BCC [2]: a Bayesian co-clustering method that enables a mixed membership in column-clusters and row-clusters.
- SOBG [27]: a co-clustering method that seeks to learn a structured optimal bipartite graph.
- ALSLCC [20]: an adaptive local structure learning method for document co-clustering.
- SCM [34]: a semi-supervised SCC for solving the problems of sparse data and noise in co-clustering.
- SSNMFCC [9]: a semi-supervised NMF method for co-clustering, in which relational matrices are computed through simultaneous modality selection and distance metric learning.
- SNCC [26]: a sparse neighbor constrained co-clustering for alleviating the misclassification of close instances.
- CoCE [39]: a co-clustering ensemble approach for learning robust co-clusters by combining multiple base co-clusterings.

Classic methods SCC, ITCC, BCC and SCM are widely used as baseline co-clustering algorithms. Other state-of-the-art methods SOBG, ALSLCC, SSNMFCC, SNCC, CoCE are either extensions of these classical methods or are most closely related to our work.

In unsupervised CALCC-KM, chunklets are obtained by performing the k-means clustering, and in order to prove the proposed two-step CALCC framework can yield better clustering performances than that of the one-step clustering approach, k-means is also introduced and compared with other co-clustering algorithms. For each data set, clustering results are obtained by applying the k-means algorithm to sample representations learned by each co-clustering algorithm.

Furthermore, in our experiments, we set the number of sample clusters k as equal to the number of feature clusters l. The number of sample chunklets $\tilde{k}$ was also set to be the same as the number of feature chunklets $\tilde{l}$. Particularly, we let $\tilde{k} = 4 \times k$ (i.e., $\tilde{l} = 4 \times l$) in our experiments. The default percentage of pairwise constraints is 5% for semi-supervised co-clustering methods. For each parameter setting, we repeated the experiments 20 times and recorded the average result for comparison.

Evaluation Metric

It is a commonly used method for measuring co-clustering quality by comparing the row (sample) clustering quality or column (feature) clustering quality between different co-clustering algorithms [19, 20, 34]. For the evaluation, we used clustering accuracy (ACC) to measure the experimental results. ACC is a widely used standard measure for clustering [6]. It measures the frequency with which all data points from the same class label reside in the same cluster [33]. ACC is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \psi(y_i, map(\hat{y}_i))}{n} \tag{18}$$

where $\psi(x, y)$ equals 1 if $x = y$ and 0 otherwise, n is the number of instances of data set X, and y_i and $\hat{y}_i$ are the true label and predicted label corresponding to instance x_i, respectively. Additionally, $map(\hat{y}_i)$ is the permutation mapping function that changes predicted labels to match the true labels by using the Kuhn–Munkres algorithm.

Table 2 Accuracy results of the experiment

Data set	K-means	ITCC	BCC	SCC	SOBG	ALSLCC	SCM	SSNMFCC	SNCC	CoCE	CALCC-KM
MM01	0.6087	0.6204	0.6080	0.6409	0.6430	0.6623	0.6450	0.5597	0.7216	0.6552	0.7469
MM02	0.4909	0.4807	0.5324	0.5570	0.5648	0.5909	0.5670	0.3991	0.6118	0.6053	0.5917
MM03	0.6501	0.6273	0.7583	0.5000	0.5009	0.7511	0.5280	0.5145	0.7218	0.7119	0.7673
MM04	0.3052	0.3655	0.4273	0.3087	0.4298	0.4202	0.3255	0.2730	0.3754	0.4309	0.4089
MM05	0.5631	0.5207	0.6381	0.6875	0.6942	0.6582	0.6877	0.5346	0.6029	0.6527	0.7005
MM06	0.3540	0.3855	0.5787	0.3520	0.3526	0.5642	0.3871	0.3512	0.5603	0.5976	0.5521
MM07	0.4422	0.4700	0.5177	0.4530	0.4530	0.5391	0.4557	0.3607	0.5471	0.5183	0.5682
MM08	0.4554	0.4276	0.5026	0.5129	0.5095	0.5577	0.5210	0.3823	0.5221	0.5413	0.5703
MM09	0.4991	0.5034	0.4707	0.5716	0.5670	0.5715	0.6021	0.3989	0.6196	0.5825	0.6603
MM10	0.3043	0.3767	0.3858	0.3946	0.5549	0.4094	0.4192	0.3901	0.4507	0.4748	0.4354
Noma	0.7105	0.7130	0.7059	0.7307	0.7130	0.7316	0.7293	0.6965	0.6993	0.7340	0.7366
Huma	0.5536	0.4841	0.5232	0.4841	0.4905	0.5049	0.4936	0.5409	0.5664	0.5829	0.5602
Av.	0.4940	0.4979	0.5541	0.5161	0.5394	0.5801	0.5301	0.4501	0.5874	0.5906	0.6082

Results

The clustering results of all 11 methods on 12 data sets are shown in Table 2. It is obvious that CALCC-KM outperforms 10 other related methods with the best average ACC of 0.6082. We observe that compared with other clustering algorithms, CALCC-KM achieves the best clustering results most of the time. In specific terms, among the 11 algorithms, CALCC-KM achieves the 7 best clustering results on the 12 data sets, while the other 10 methods only achieve 5 of the best clustering results. Another important observation is that SCM achieves a better clustering quality than SCC, which verifies the assumption that prior information can reduce the noise and effectively enhance the clustering performance. We also observe that on average, most of co-clustering methods outperform the *k*-means clustering, this is because co-clustering algorithms can utilize the relations between sample clusters and feature clusters, as a consequence, they make the data sets more predictable and the co-clustering performance more excellent compared with traditional one-side clustering.

The Friedman-aligned test [19] is introduced to make a further comparison among those algorithms. Table 3 shows the aligned observations and the aligned ranks in the parentheses with consideration of the known 11 algorithms and 12 data sets. As shown in the table, on average, CALCC-KM ranks first at 24.58. The Friedman aligned test can be used to check whether the measured sum of aligned ranks is different from the total aligned ranks $\hat{R}_j = 731$ at the high level of significance expected under the null hypothesis:

$$\sum_{j=1}^{n} \hat{R}_{i,\cdot}^2 = 767^2 + 699^2 + \ldots + 711^2 + 755^2 = 6,429,978$$

$$\sum_{j=1}^{k} \hat{R}_{\cdot,j}^2 = 1172^2 + 1164^2 + \ldots + 407^2 + 295^2 = 8,224,698$$

$$T = \frac{(11-1)[8,224,698 - (11 \cdot 12^2/4)(11 \cdot 12+1)^2]}{11 \times 12(11 \times 12+1)(2 \times 11 \times 12+1)/6 - 6,429,978/11} = 59.68$$

With 11 algorithms and 12 data sets, T is distributed according to the Chi-square distribution with $11-1=10$ degrees of freedom. The p-value computed by using the $\chi^2(10)$ distribution is 0.00000025; thus, the null hypothesis is significantly rejected. It is obvious that the value is far less than 0.05, which shows that the results of the algorithms are significantly different.

Unsupervised CALCC-KM versus Semi-Supervised Co-Clustering

To further illustrate the superior performance of the proposed framework, for each of the 12 data sets, we plot the average

Table 3 Aligned observations of 11 algorithms selected in the experimental study. Ranks in parentheses are used in the computation of the Friedman aligned ranks test

Data set	K-means	ITCC	BCC	SCC	SOBG	ALSLCC	SCM	SSNMFCC	SNCC	CoCE	CALCC-KM	Total
MM01	-0.3782(97)	-0.2612(88))	-0.3852(98)	-0.0562(77)	-0.0352(73)	0.1578(57)	-0.0152(71)	-0.8682(116)	0.7508(14)	0.0868(66)	1.0038(10)	767
MM02	-0.5379(107)	-0.6399(109)	-0.1229(81))	0.1231(62)	0.2011(52)	0.4621(35)	0.2231(46)	-1.4559(131)	0.6711(19)	0.6061(23)	0.4701(34)	699
MM03	0.109(64)	-0.119(80)	1.191(5)	-1.392(130)	-1.383(129)	1.119(6)	-1.112(124)	-1.247(128))	0.826(13)	0.727(15)	1.281(3)	697
MM04	-0.6484(110)	-0.0454(76)	0.5726(26)	-0.6134(108)	0.5976(24)	0.5016(33)	-0.4454(103)	-0.9704(118)	0.0536(68)	0.6086(22)	0.3886(38)	726
MM05	-0.6783(111)	-1.1023(123)	0.0717(67)	0.5657(30)	0.6327(20)	0.2727(45)	0.5677(29)	-0.9633(117)	-0.2803(90)	0.2177(50)	0.6957(18)	700
MM06	-1.0375(119)	-0.7225(113)	1.2095(4)	-1.0575(121)	-1.0515(120)	1.0645(8)	-0.7065(112)	-1.0655(122)	1.0255(9)	1.3985(1)	0.9435(11)	740
MM07	-0.4189(102)	-0.1409(82)	0.3361(41)	-0.3109(92.5)	-0.3109(92.5)	0.5501(31)	-0.2839(91)	-1.2339(127))	0.6301(21)	0.3421(40)	0.8411(12)	732
MM08	-0.4485(104)	-0.7265(114)	0.0235(69)	0.1265(61)	0.0925(65)	0.5745(25)	0.2075(51)	-1.1795(126)	0.2185(48)	0.4105(36)	0.7005(16)	715
MM09	-0.506(106)	-0.463(105)	-0.79(115)	0.219(47)	0.173(55)	0.218(49)	0.524(32)	-1.508(132)	0.699(17)	0.328(43)	1.106(7)	708
MM10	-1.1351(125)	-0.4111(99)	-0.3201(94)	-0.2321(87)	1.3709(2)	-0.0841(78)	0.0139(70)	-0.2771(89)	0.3289(42)	0.5699(28)	0.1759(54)	768
Noma	-0.159(83)	-0.044(74.5)	-0.115(79)	0.133(60)	-0.044(74.5)	0.142(59)	0.119(63)	-0.209(85)	-0.181(84)	0.166(56)	0.192(53)	771
Huma	0.2775(44)	-0.4175(100.5)	-0.0265(72)	-0.4175(100.5)	-0.3535(96)	-0.2095(86)	-0.3225(95)	0.1505(58)	0.4055(37)	0.5705(27)	0.3435(39)	755
Total	1172	1164	751	976	803	512	887	1349	462	407	295	
Av.	97.67	97.00	62.58	81.33	66.92	42.67	73.92	112.42	38.50	33.92	24.58	

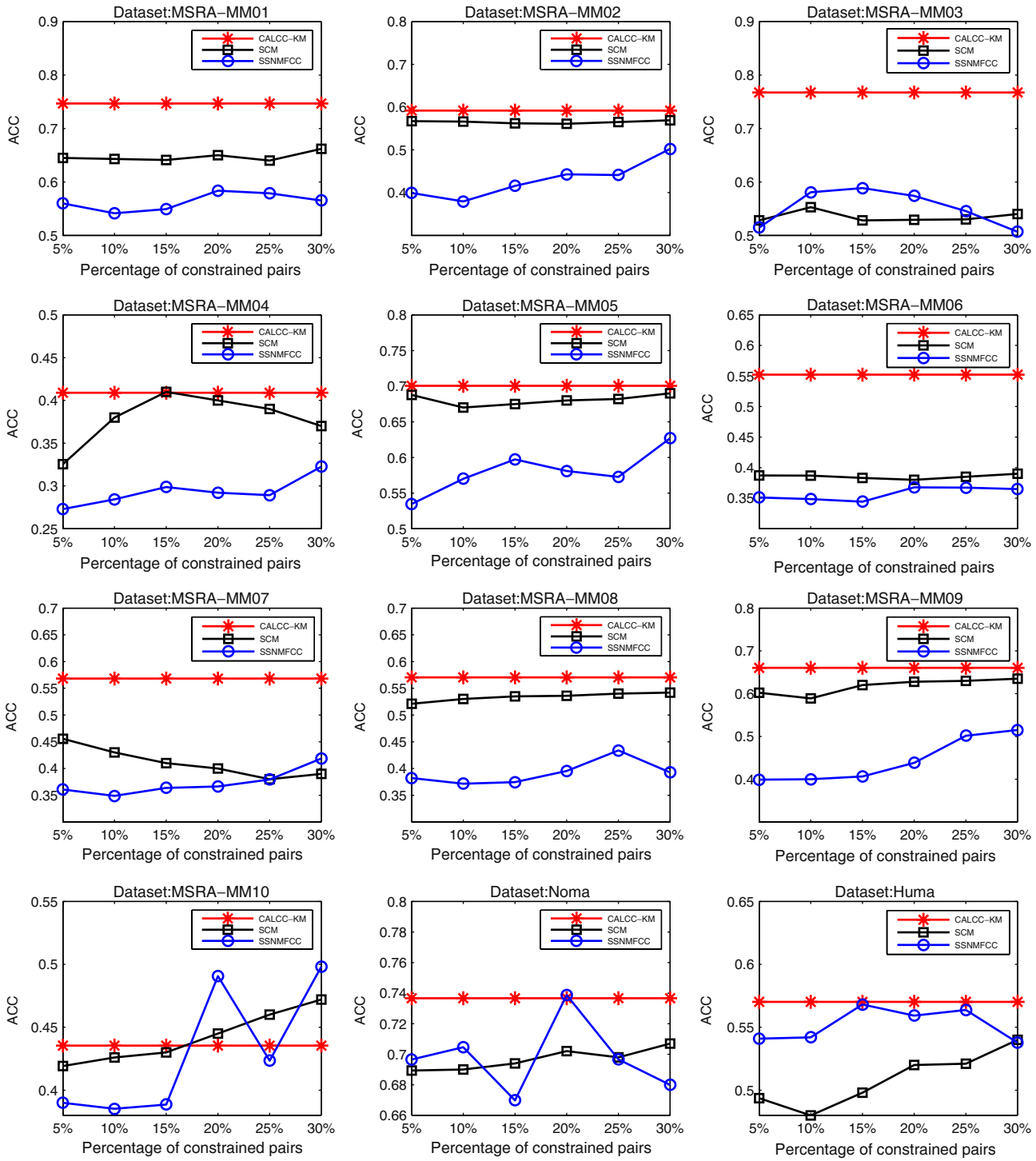

Fig. 2 Experimental results for unsupervised co-clustering CALCC-KM and semi-supervised co-clustering SCM and SSNMFCC on 12 data sets against the increasing percentage of pairwise constraints

ACC value against the increasing percentage of pairwise constraints for our unsupervised CALCC-KM algorithm and two other semi-supervised SSNMFCC and SCM algorithms. Since our CALCC-KM is an unsupervised co-clustering algorithm, the accuracy of CALCC-KM is not affected by the percentage of pairwise constraints and is always a constant. The ACC results are shown in Figs. 2 and 3. Surprisingly, as observed in Fig. 2, our unsupervised CALCC-KM significantly outperforms the other two semi-supervised SCM and SSNMFCC methods in most cases. In Fig. 3, it is obvious

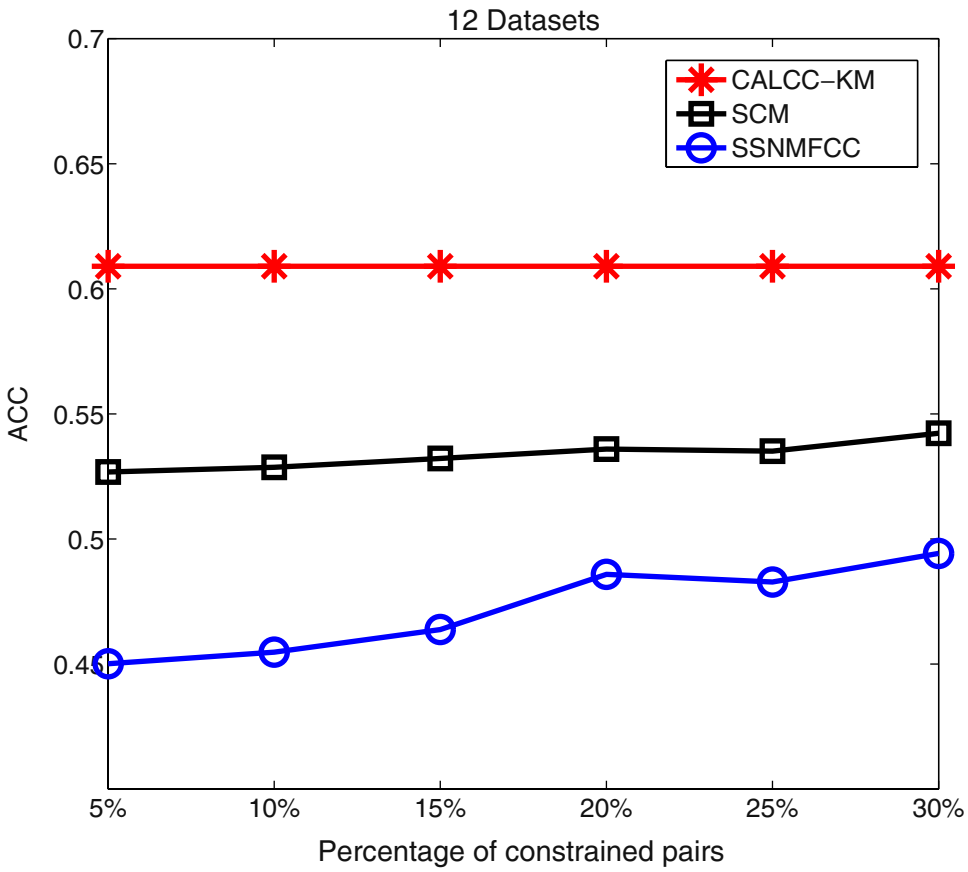

Fig. 3 Average ACC of 12 data sets for unsupervised co-clustering CALCC-KM and semi-supervised co-clustering SCM and SSNMFCC against the increasing percentage of pairwise constraints

that the average ACC of CALCC-KM on 12 data sets is close to 61%. It is almost 8% and 20% higher than that of SCM and SSNMFCC, respectively. Furthermore, the average accuracies of SCM and SSNMFCC consistently increase with the gradual increase of the pairwise constraint percent (from 5 to 30 percent). Particularly, the ACC curve of SCM rises relatively gently as the pairwise constraints increase, which may be caused by the problem of constraint conflicts. In general, as an unsupervised co-clustering algorithm, CALCC-KM can generate far superior clustering quality than some of related semi-supervised co-clustering algorithms.

Two-Side CALCC versus Its One-Side Counterpart

The proposed CALCC is a two-side co-clustering framework. In this framework, not only chunklets from sample side, but also chunklets from feature side are used to guide the learning process of the CAL model. In order to prove the availability of the information extracted from those feature chunklets, we evaluate the clustering performance of one-side CALCC with only minimizing the sample chunklet scatter values. In this case, the objective function (5) is reduced to $J(U) = S_s$. And the sample-projective matrix U can be easily obtained according to equations (10)-(17) (by simply setting $V = U$).

A novel one-side clustering method named CALCC-KM1 was proposed to make a comparison with its two-side counterpart CALCC-KM. K-means clustering was used to generate sample chunklets from the original data set in CALCC-KM1. The clustering results of the k-means, CALCC-KM and CALCC-KM1 algorithms over 12 data sets are shown in Table 4.

Table 4 The clustering results of the K-means, CALCC-KM and CALCC-KM1 algorithms over 12 MSRA-MM and UCI data sets. Note that the highest score is denoted by bold font, and the second is underlined

Methods	MM01	MM02	MM03	MM04	MM05	MM06	MM07	MM08	MM09	MM10	Noma	Huma	Av.
k-means	0.6087	0.4909	0.6501	0.3052	0.5631	0.3540	0.4422	0.4554	0.4991	0.3043	0.7105	0.5536	0.4940
CALCC-KM1	0.5773	0.5481	0.6926	**0.4118**	0.6603	0.4008	0.4159	**0.5829**	0.4855	0.3972	0.6943	**0.5883**	0.5379
CALCC-KM	**0.7469**	**0.5917**	**0.7673**	0.4089	**0.7005**	**0.5521**	**0.5682**	0.5703	**0.6603**	**0.4354**	**0.7366**	0.5602	**0.6082**

The table shows that two-side CALCC can achieve better performance than its one-side counterpart, supporting the view that two-side co-clustering can obtain better insight into the data and make the input data more predictable than traditional one-side clustering. We also notice that the use of sample chunklets can improve the results as well, which also verifies the effectiveness of the chunklet information.

Discussion on Chunklet Number

In this subsection, we describe our empirical evaluation of the impact of different numbers of chunklets on the performance of unsupervised CALCC. Considering the different characteristics of the original data matrix and the experiment effectiveness, we represent different numbers of chunklets as different multiples of cluster numbers. The parameter α is used to represent multiples of the cluster number, i.e., $\tilde{k} = \alpha \times k$, $\tilde{l} = \alpha \times l$. In our experiments, the value of α was varied from one to ten in steps of one. The clustering results on 12 data sets are plotted in Fig. 4. As can be seen from the figure, CALCC-KM achieves varying clustering quality at different α values for different data sets. To summarize, data sets with different distributions may be composed of different numbers of chunklets that can easily find boundaries.

Significantly, for the simplicity of the experiments, we only considered pursuing the optimal α in the case of $\alpha \leq 10$. Thus, the optimal α obtained is actually the local optimal α. However, the experimental results show that even in the case of the local optimum, our CALCC-KM has achieved the best results over the other related co-clustering algorithms. This means that a better parameter setting can help achieve a better clustering quality than that reported in this paper.

Effectiveness of Our Model with Different Modalities

Since the proposed unsupervised CALCC is formulated for more than one modality, we evaluated its effectiveness in the three modalities on the last 7 data sets in Table 1. In addition to CALCC-KM, we introduce two other modalities, CALCC-DP and CALCC-AP. More specifically, CALCC-DP and CALCC-AP indicate that density peaks (DP) [30] and AP clustering are, respectively, applied to generate chunklets on sample and feature dimensions. The results are shown in Fig. 5. We can observe that all of three modalities achieve a high cluster quality over the seven data sets. In addition, CALCC-DP outperforms the other two modalities in Spect and Credi data sets, and CALCC-AP achieves the best performance in Ionosphere, Sona, Vote, and Wdbc data sets. However, only in the Syncon data set does CALCC-KM obtain the highest accuracy. The reason for that might be because compared with DP and AP clustering *k*-means is not the most suitable clustering method to generate clearly structured chunklets on the other six data sets. In general terms, in unsupervised CALCC, different unsupervised clustering algorithms for chunklet generation can achieve significantly different co-clustering performances in accordance with the different data characteristics.

Study on Time Complexity

The time complexity of the CAL model is mainly determined by the singular value decomposition (SVD) process of the covariance matrix W in Eq. (11). The matrix W is a square matrix of order $n + p$ for a data matrix $\mathbf{X}_{n \times p}$ with n rows and p columns. It is known that exact SVD of a q-order matrix has time complexity $O(q^3)$, so the time complexity of the model is $O((n + p)^3)$. For the proposed

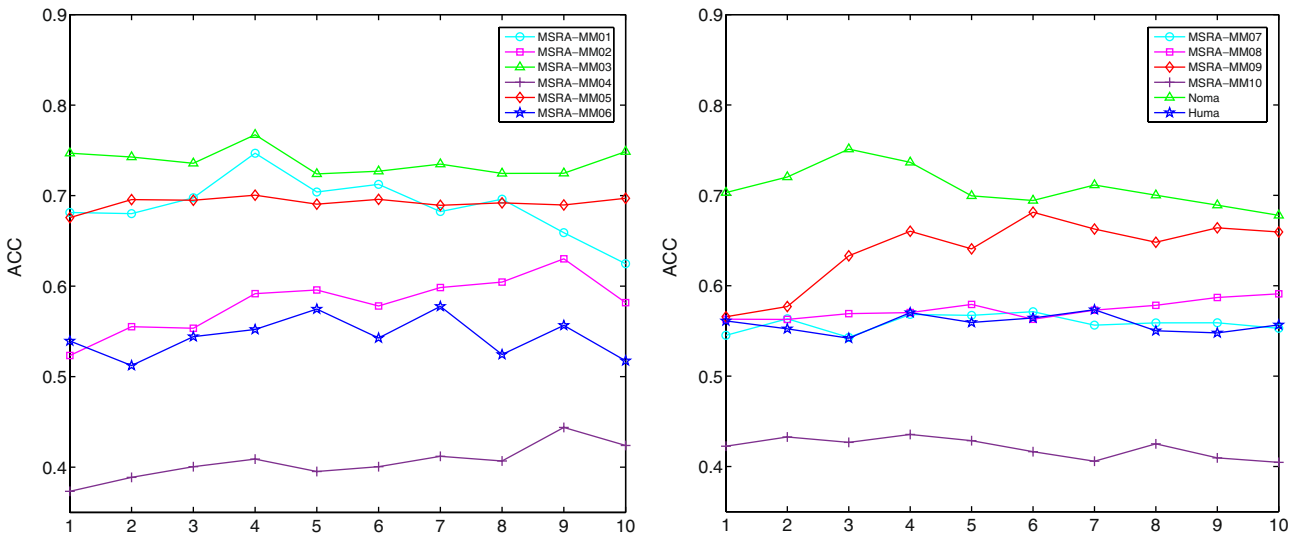

Fig. 4 Cluster performance of all 12 data sets with different cluster-number multiples

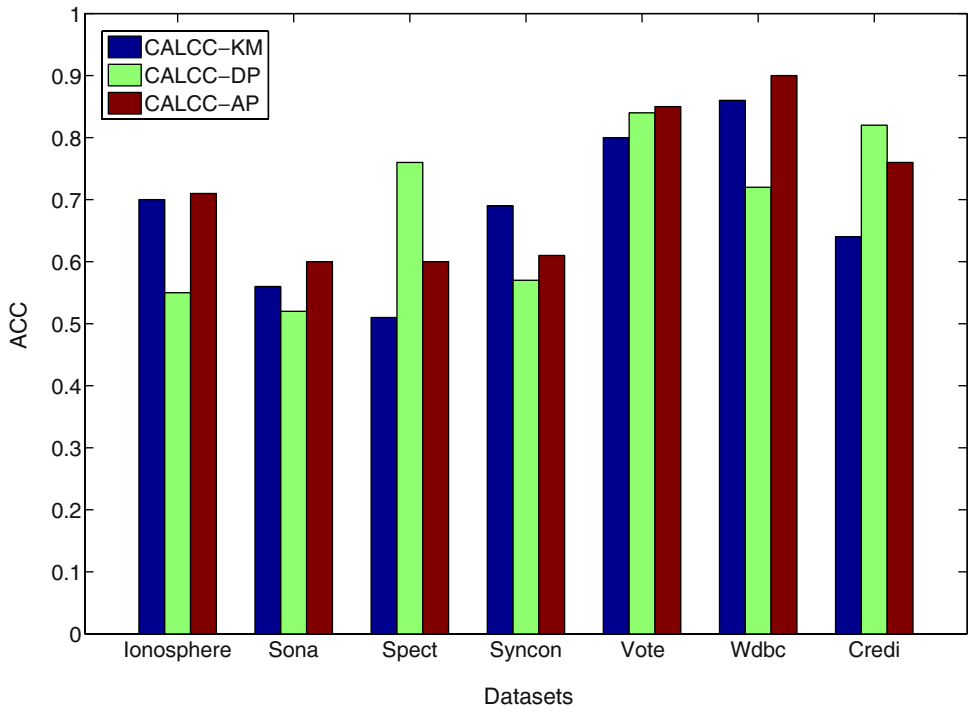

Fig. 5 Experimental results for three CALCC modalities CALCC-KM, CALCC-DP and CALCC-AP over 7 UCI data sets

CALCC-KM algorithm, it costs another $O(n\tilde{k}g_s + p\tilde{l}g_f)$ time complexity to construct chunklets and another $O(nkc_s + plc_f)$ time complexity to obtain clustering results, where $\tilde{k}$ (k), $\tilde{l}$ (l) are the number of row-chunklets (row-clusters) and column-chunklets (column-clusters), respectively; g_s (g_f) is the number of sample (feature) chunklet generation iterations, c_s (c_f) is the number of sample (feature) clustering iterations.

Conclusion

In this paper, a novel co-clustering framework named co-adjustment learning for co-clustering (CALCC) was proposed. The proposed CALCC can be flexibly used in unsupervised, semi-supervised and supervised learning situations. A constraint co-projection model, co-adjustment learning (CAL), was first introduced. The CAL model not only makes full use of the informative chunklet constraints, but also transforms the original data into a new discriminative representation space by simultaneously performing sample projection as well as feature projection. In the transformed space, the co-clusters with lowest total variance can be efficiently found. To the best of our knowledge, the presented CAL model that exploits constraint information from chunklets for co-projection is the first to be introduced. In order to prove the availability of our framework, an unsupervised case of CALCC was introduced to make an extensive comparison with several related co-clustering methods on several image and real data sets. Besides, we also performed a comparative experiment on the three modalities of unsupervised CALCC. The experimental results revealed the superior performance of the CAL model in discovering discriminative representations and demonstrated the effectiveness of the proposed framework.

Acknowledgements This work is partially supported by Key program for International S&T Cooperation of Sichuan Province, No. (2019YFH0097); Science and Technology Support Project of Sichuan Province under290 Grant No. 2020YFG0045, 2020YFG0238.

Compliance with Ethical Standards

Conflicts of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abdullah A, Hussain A. A cognitively inspired approach to two-way cluster extraction from one-way clustered data. Cogn Comput. 2014;7(1):161–82.

2. Banerjee A, Dhillon I, Ghosh J, Merugu S, Modha DS. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. J Mach Learn Res. (Aug) 2007;8:1919–86.
3. Bekkerman R, Ran EY, Mccallum A. Multi-way distributional clustering via pairwise interactions. International Conference on Machine Learning. 2005:41–8.
4. Blake C. Uci repository of machine learning databases. Department of Information and Computer Science: University of California; 1998.
5. Busygin S. Biclustering in data mining. Computers and Operations Research. 2008;35(9):2964–87.
6. Cai D, He X, Han J, Huang TS. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell. 2011;33(8):1548–60.
7. Chen LC, Yu PS, Tseng VS. Wf-msb: a weighted fuzzy-based biclustering method for gene expression data. Int J Data Min Bioin. 2011;5(1):89–109.
8. Chen, X., Ritter, A., Gupta, A., Mitchell, T.: Sense discovery via co-clustering on images and text. In: IEEE Conference on Computer Vision and Pattern Recognition. 2015:5298–5306.
9. Chen Y, Wang L, Dong M. Non-negative matrix factorization for semisupervised heterogeneous data coclustering. IEEE Transactions on Knowledge Data Engineering. 2010;22(10):1459–74.
10. Cheng Y, Church GM. Biclustering of expression data. In: Eighth International Conference on Intelligent Systems for Molecular Biology. 2000:93–103.
11. Dhillon I, Mallela S, Modha D. Information-theoretic co-clustering. In: ACM SIGKDD International Conference on Knowledge Discovery Data Mining. 2003:89–98.
12. Dhillon IS. Co-clustering documents and words using bipartite spectral graph partitioning. In: ACM SIGKDD International Conference on Knowledge Discovery Data Mining. 2001:269–274.
13. Feldman DD, Griffiths LJ. A constraint projection approach for robust adaptive beamforming. In: Acoustics, Speech, and Signal Processing, 1991. Icassp-91., 1991 International Conference. 1991:1381–84.
14. Gao C, Mcdowell IC, Zhao S, Brown CD, Engelhardt BE. Context specific and differential gene co-expression networks via bayesian biclustering: Plos Computational Biology. 2016;12(7):e1004791.
15. Garcia-Pedrajas N, Maudes-Raedo J, Garcia-Osorio C, Rodriguez-Diez JJ. Supervised subspace projections for constructing ensembles of classifiers. Inf Sci. 2012;193(11):1–21.
16. Hartigan JA. Direct clustering of a data matrix. Publications of the American Statistical Association. 1978;67(337):123–9.
17. Herlocker JL, Konstan JA, Terveen L, Riedl J. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems. 2004;22(1):5–53.
18. Huang S, Wang H, Li T, Li T, Xu Z. Robust graph regularized nonnegative matrix factorization for clustering. Data Mining and Knowledge Discovery. 2018;32(2):483–503.
19. Huang S, Wang H, Li T, Yang Y, Li T. Constraint co-projections for semi-supervised co-clustering. IEEE Transactions on Cybernetics. 2015;46(12):3047–58.
20. Huang S, Xu Z, Lv J. Adaptive local structure learning for document co-clustering. Knowledge-Based Systems. 2018;148:74–84.
21. Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Computing Surveys. 1999;31(3):264–32323.
22. Klein D, Kamvar SD, Manning CD. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of the Nineteenth International Conference on Machine Learning. 2020:307–314.
23. Kumar S, Gao X, Welch I. Learning under data shift for domain adaptation: A model-based co-clustering transfer learning solution. In: Pacific Rim Knowledge Acquisition Workshop. 2016:43–54.
24. Leung WT, Lee DL, Lee WC. Clr:a collaborative location recommendation framework based on co-clustering. 2011:305–14.
25. Li H, Wang M, Hua XS. Msra-mm 2.0: A large-scale web multimedia dataset. In: IEEE International Conference on Data Mining Workshops. 2009:164–69.
26. Lu Z, Liu G, Wang S. Sparse neighbor constrained co-clustering via category consistency learning. Knowledge-Based Systems. 2020.
27. Nie F, Wang X, Deng C, Huang H. Learning a structured optimal bipartite graph for co-clustering. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds.) Advances in Neural Information Processing Systems 30. 2017:4129–38. Curran Associates, Inc. http://papers.nips.cc/paper/7001-learning-a-structured-optimal-bipartite-graph-for-co-clustering.pdf
28. Peng X, Xu D. Structural regularized projection twin support vector machine for data classification. Information Sciences. 2014;279(279):416–32.
29. Prats-Montalbn JM, Lopez F, Valiente JM, Ferrer A. Feature extraction and classification in surface grading application using multivariate statistical projection models. In: Eighth International Conference on Quality Control by Artificial Vision. 2007:63560N–63560N–11.
30. Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Machine Learning. 2014;344(6191):1492.
31. Shan H, Banerjee A. Bayesian co-clustering. In: Eighth IEEE International Conference on Data Mining. 2008:530–39.
32. Shental N, Bar-Hillel A, Hertz T, Weinshall D. Gaussian mixture models with equivalence constraints. Constrained Clustering. 2009:33–58.
33. Shental N, Hertz T, Weinshall D, Pavel M. Adjustment learning and relevant component analysis. In: European Conference on Computer Vision. 2002:776–92.
34. Shi, X, Fan W, Yu PS. Efficient semi-supervised spectral co-clustering with constraints. In: IEEE International Conference on Data Mining. 2011:043–48.
35. Wang P, Domeniconi C, Laskey KB. Nonparametric bayesian clustering ensembles. In: European Conference on Machine Learning and Knowledge Discovery in Databases. 2010:435–50.
36. Webb AR, Copsey KD. Introduction to statistical pattern recognition. Academic Press, 1972:2133–43
37. Whang JJ, Dhillon IS. Non-exhaustive, overlapping co-clustering. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017;2367–70.
38. Yan Y, Chen L, Tjhi WC. Fuzzy semi-supervised co-clustering for text documents. Fuzzy Sets and Systems. 2013;215(215):74–89.
39. Yu X, Yu G, Wang J, Domeniconi C. Co-clustering ensembles based on multiple relevance measures. IEEE Transactions on Knowledge and Data Engineering. 2019.
40. Zhang, Daoqiang, Chen, Songcan, Zhou, ZhiHua, Yang, Qiang. Constraint projections for ensemble learning. National conference on artificial intelligence. 2008;758–763.