Knowl Inf Syst (2018) 56:257–284
https://doi.org/10.1007/s10115-017-1110-9

SURVEY PAPER

Subspace multi-clustering: a review

Juhua Hu[1] · **Jian Pei**[1]

Received: 01 April 2017 / Revised: 18 July 2017 / Accepted: 30 July 2017 /
Published online: 4 October 2017

Abstract Clustering has been widely used to identify possible structures in data and help users to understand data in an unsupervised manner. Traditional clustering methods often provide a single partitioning of the data that groups similar data objects in one group while separates dissimilar ones into different groups. However, it has been well recognized that assuming only a single clustering for a data set can be too strict and cannot capture the diversity in applications. Multiple clustering structures can be hidden in the data, and each represents a unique perspective of the data. Different multi-clustering methods, which aim to discover multiple independent structures hidden in data, have been proposed in the recent decade. Although multi-clustering methods may provide more information for users, it is still challenging for users to efficiently and effectively understand each clustering structure. Subspace multi-clustering methods address this challenge by providing each clustering a feature subspace. Moreover, most subspace multi-clustering methods are especially scalable for high-dimensional data, which has become more and more popular in real applications due to the advances of big data technologies. In this paper, we focus on the subject of subspace multi-clustering, which has not been reviewed by any previous survey. We formulate the subspace multi-clustering problem and categorize the methodologies in different perspectives (e.g., de-coupled methods and coupled methods). We compare different methods on a series of specific properties (e.g., input parameters and different kinds of subspaces) and analyze the advantages and disadvantages. We also discuss several interesting and meaningful future directions.

Keywords Multi-clustering · Subspace multi-clustering · De-coupled · Coupled

This work is supported in part by the NSERC Discovery Grant program, the Canada Research Chair program, and the NSERC Strategic Grant program. All opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

✉ Juhua Hu
juhuah@sfu.ca

[1] School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

1 Introduction

Clustering, also known as cluster analysis, aims to group a set of objects in such a way that objects in the same cluster are more similar to each other than to those in other clusters [40]. It has been widely used to discover hidden structures in data. Taking computational genomic as an example, clustering can be used to build groups of genes with related expression patterns and such groups contain functionally related proteins, such as enzymes for a specific pathway [11].

Many clustering methods were proposed in the literature. Some recent surveys on the subject include [12,40,72,73]. Traditional clustering methods [40] focus on finding a single way to partition data into groups. However, it has been well recognized that different outputs are possible if one varies the parameter setting, changes the clustering algorithm, or uses different subsets of features in analysis. This means that assuming only a single clustering for a data set can be too strict, which has motivated the emerging area of multi-clustering [55].

Due to the advances of data storage and data collection techniques, it has become more and more necessary and natural to assume multiple clustering structures over a data set. For example, more and more features can be collected for a set of objects. These features may be of different types (e.g., text, image, or audio) or from multiple sources (e.g., customer behaviors from multiple markets such as financial investment, vacation expenditure, and entertainment expense). Different combinations of these features may provide orthogonal ways to partition the objects, each way presenting a unique perspective. An illustrative example is shown in Fig. 1. Given a set of data points whose attributes are their colors and shapes as shown in Fig. 1a, there exist two orthogonal ways to group them: Fig. 1b grouping by their color feature and Fig. 1c by their shape feature.

To obtain multiple good and different clusterings, two general strategies were proposed. Semi-supervised multi-clustering methods [10,21,75] focus on finding one or more alternative clusterings with respect to a given clustering. They generate multiple clusterings in a greedy way such that multiple clusterings are produced sequentially and a new clustering is required to be different from the previous ones. Unsupervised multi-clustering methods [16,23,41] try to simultaneously generate multiple clusterings that are constrained to be different from each other.

Apparently, multi-clustering methods provide more information for users, but also lead to a challenge to users: how to efficiently and effectively understand many clusterings. Subspace multi-clustering methods provide each clustering a corresponding feature subspace. Users can interpret and understand each clustering structure using its feature subspace, such as color or shape in the example of Fig. 1. Moreover, subspace multi-clustering methods are in a great demand for high-dimensional data that presents a major challenge in the big data era.

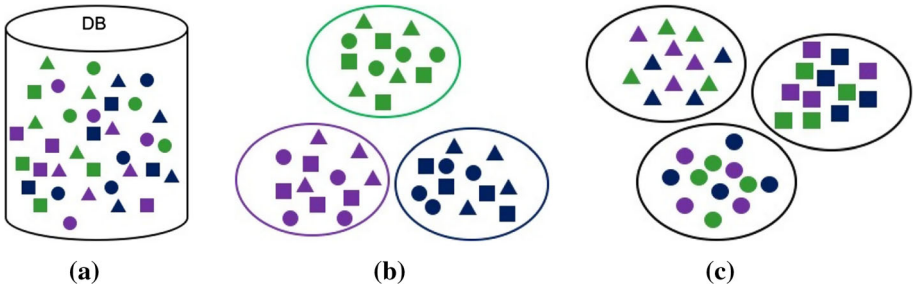

Fig. 1 An example illustrating the idea of multi-clustering. **a** Original data, **b** grouping by color, **c** grouping by shape

1.1 Applications

More often than not, multiple phenotypes are hidden in data. It is unlikely a single clustering will meet all users' interests. Subspace multi-clustering methods are applicable in a board range of real-world tasks especially for high-dimensional data. In this subsection, we present some specific application scenarios as examples where subspace multi-clustering methods are potentially helpful.

1.1.1 Content-based image retrieval

Content-based image retrieval (CBIR) [51] aims to retrieve images based on their content that refers to colors, shapes, textures, or any other information that can be derived from the image itself. In a typical CBIR setting, a user poses a query (e.g., an example image) and asks the system to bring out relevant images from the database. The most common method is to compare the query image with all images from the database using an image similarity measure [69]. The main challenge here is that an image can be perceived with different meanings, and thus, the similarity between the same pair of images may change when the concept being queried changes.

The query concepts are usually hidden in different subspaces. Subspace multi-clustering methods can be applied to the database to discover different subspaces, each corresponding to a clustering structure over the database. Thereafter, when a query image comes, different set of relevant images from the database can be output in different subspaces, each corresponding to one potential query concept. For example, a CBIR system may contain a database of fruit images. Given a query image, a user may want similar fruits in different concepts, e.g., fruits with similar shapes, fruits with similar colors, and fruits in the same specie. Hu et al. [36,37] applied a multiple stable clustering method to a fruit data set and found different partitions as shown in Fig. 2.

In some scenarios, users may provide some relevance feedback by marking images in the results as *not relevant* to the search query. Some semi-supervised subspace multi-clustering methods can use the feedback to discover alternative subspaces, and then refine the result.

1.1.2 Customer relationship management

Customer relationship management (CRM) [48] is a process used by companies to understand their customer groups, where customer segmentation aims to discover customer groups that share similar characteristics to foster customers' full profit potential. Due to the advances of big data technologies, the profiles of customers are being continuously collected and each

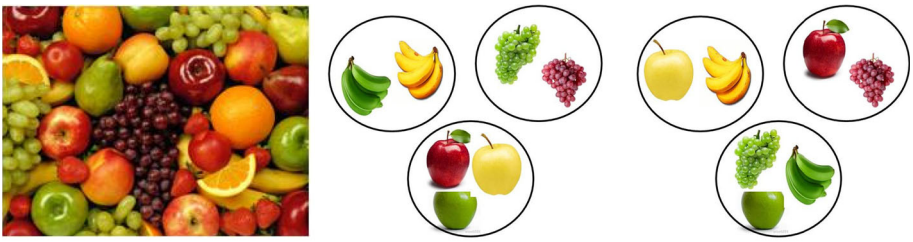

Fig. 2 An example of multi-clustering in CBIR [36,37]. **a** Fruits, **b** clustering by species, **c** clustering by color

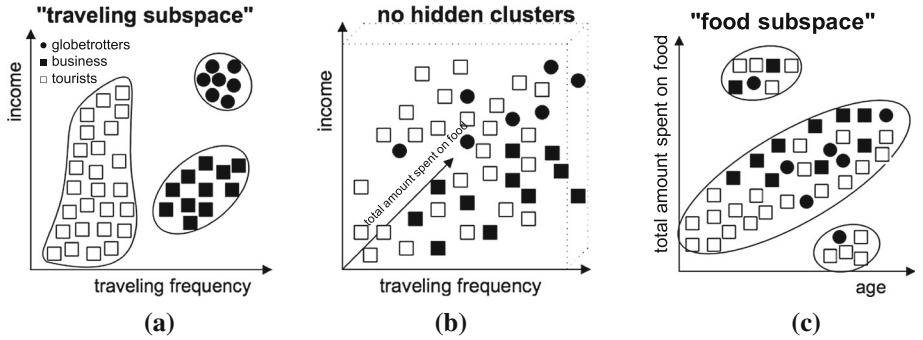

Fig. 3 An example of customer segmentation in different subspaces [55]

customer may be described by a series of attributes such as customer id, age, gender, and income.

Clustering on customer profiles can help derive groups of customers with similar interests. This group information is very useful in product recommendations. For example, similar travelers as shown in Fig. 3a may buy similar travel packages, where "traveling subspace" is a subspace composed by two attributes, that is, traveling frequency and income. However, when there are many attributes, customer data may be sparse, that is, very few customers are similar to each other on many attributes. Taking Fig. 3b as an example, if we add one irrelevant feature, total amount spent on food, to the traveling subspace, no clustering structure is observable.

Moreover, companies (e.g., retailers as Amazon or Costco) usually have a variety of products (e.g., travel packages and food products) to promote to their customers. However, customers can be grouped differently in different perspectives, e.g., customers who are similar travelers may have totally different interests on food products of this company as shown in Fig. 3c, where "food subspace" is a subspace comprised by attributes age and total amount spent on food.

In summary, there are two main challenges in customer segmentation: (1) clustering structures are hidden in unknown feature subspaces; (2) a single clustering structure may not capture end users' interests. Subspace multi-clustering methods address these two challenges by finding multiple different clustering structures in different subspaces. Then, end users can intuitively understand each clustering structure through the corresponding subspace and choose one or more clustering structures that are useful for their specific application purposes. For example, a subspace multi-clustering method may find two different clustering structures, that is, groups of customers in Fig. 3a and those in Fig. 3c, respectively. Then, end users can understand each clustering structure through the features used for the corresponding subspace. Specifically, a composition of attributes traveling frequency and income implies a traveling subspace, while that of attributes age and total amount spent on food indicates a food subspace. Thereafter, end users may choose one or both of them to do product promotions, Fig. 3a for travel packages and/or Fig. 3c for food products.

1.1.3 DNA microarray analysis

The DNA microarray technology [63] enables our understanding of complex cellular mechanisms through patterns of gene expression. Microarray data sets usually provide information about the expression levels of different genes under different conditions. For example, a lymphoma microarray data set can be represented by a gene expression matrix in which a

Genes	Conditions					
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
TNNC1	14.82	14.46	14.76	19.22	19.55	19.18
DKK4	14.71	14.37	14.23	19.74	19.73	19.78
ZNF185	14.20	14.96	14.07	19.57	19.37	19.10
CHST3	14.40	14.18	14.15	16.18	16.99	16.03
FABP3	14.87	14.80	14.85	16.16	16.99	16.05
MGST1	11.76	11.80	11.67	19.92	19.02	19.32
DEFA5	11.63	11.47	11.54	19.52	19.52	19.37
VIL1	11.47	11.69	11.87	16.94	16.01	16.72
AKAP12	11.26	11.10	11.50	16.60	16.69	16.62
HS3ST1	11.61	11.67	11.50	16.44	16.23	16.61

Fig. 4 An example of DNA microarray data (color figure online)

row represents a gene and a column represents a cancer sample (a condition) as shown in Fig. 4. Each numerical value in the matrix characterizes the expression level of a specific gene under a particular condition.

Understanding groups of genes with similar functions is helpful for finding appropriate treatments for specific conditions. However, one gene may have different functions under different conditions, that is, genes with similar functions under one condition may be different under another. For example, as shown in Fig. 4, we can cluster those genes into two groups under conditions 1 to 3, i.e., green group $=\{$TNNC1, DKK4, ZNF185, CHST3, FABP3$\}$ and orange group $=\{$MGST1, DEFA5, VIL1, AKAP12, HS3ST1$\}$. At the same time, we can group them into another two partitions under conditions 4 to 6, i.e., blue group$=\{$TNNC1, DKK4, ZNF185, MGST1, DEFA5$\}$ and gray group$=\{$CHST3, FABP3, VIL1, AKAP12, HS3ST1$\}$. Subspace multi-clustering methods are promising on finding such different group structures under different conditions.

1.1.4 Sensor surveillance

Nowadays, sensors have been widely used to collect different attributes of their environments using different measurements, e.g., the temperature and humidity. Clustering on this kind of sensors can derive environmental conditions in a large area, which is useful for weather broadcasting or forecasting [1,47].

However, regions can be grouped differently according to different criteria. For example, we may group regions in USA into clusters of high temperature, medium temperature, and low temperature according to their temperatures. Then, we can provide advises on how to avoid heatstroke for people in the high-temperature area, while reminding people in the low-temperature area to keep warm. At the same time, we may also cluster regions with different precipitations of rain and let people in different clusters pay different attentions on the rain. This is essentially a subspace multi-clustering problem.

1.1.5 Social network analysis

Due to the rapid growth of social media, especially online social networking applications such as Facebook, YouTube, LinkedIn, and Twitter, people are closely connected via different types of relationships. People belonging to a tight-knit community online are more likely to

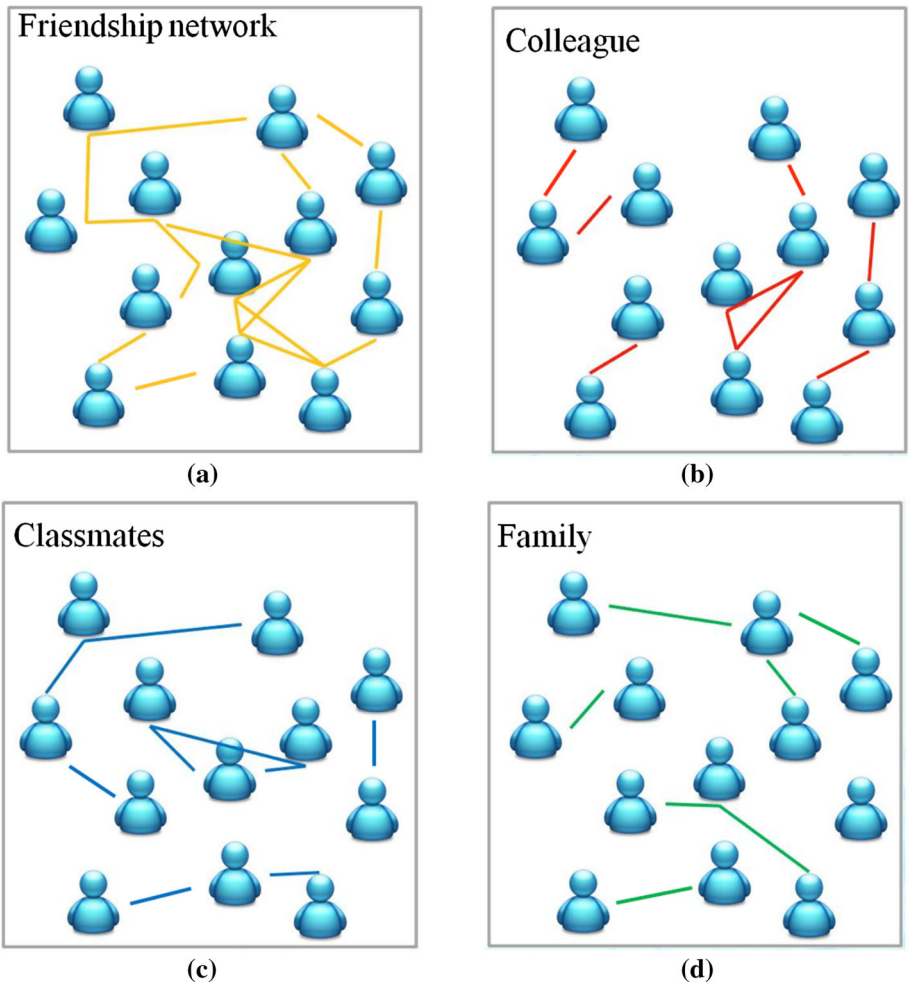

Fig. 5 An example of different communities hidden in different subspaces

have some properties in common, which can help us understand interactions between people better. The capability to find communities within large networks in some automated fashion is of considerable use. For example, people in a tight-knit Facebook community may have common political opinions [57]. Therefore, the problem of finding communities [32,50] has been the focus of many efforts in social network studies.

However, real social networks are often heterogeneous, that is, people are actually connected via different types of social ties, e.g., in a mobile communication network, the relationship types may include family, classmates, colleagues, and friends. More importantly, it is well recognized that different types of social relationships have essentially different influence between people, e.g., a person's classmates may influence his/her choices of courses to take, while his/her career path may be more affected by his/her colleagues.

As an instance, considering some persons connected by online friendship in Fig. 5a (it is possible that some friends in real world do not connect online), it is hard to tell the

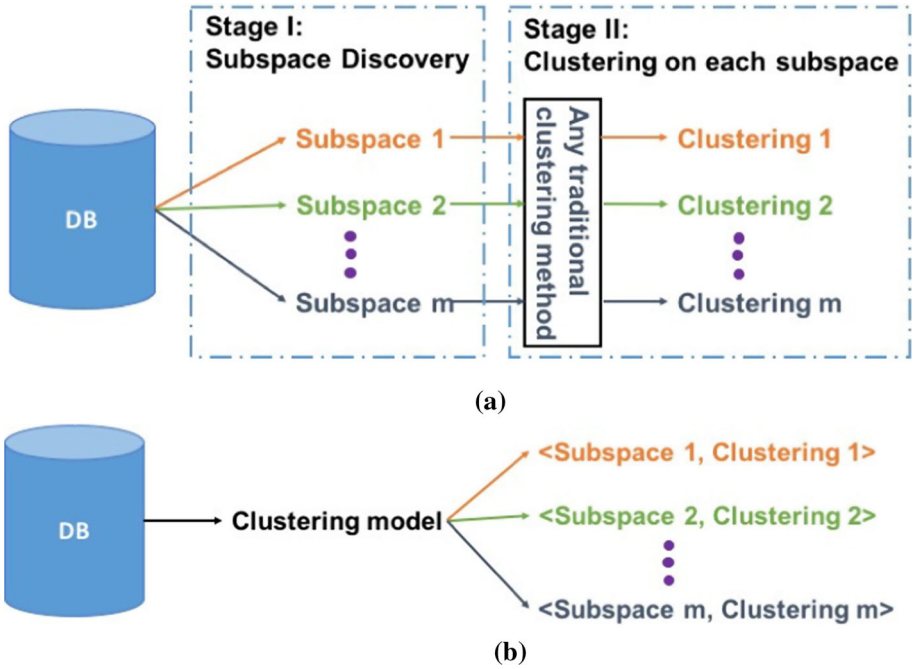

Fig. 6 Two types of subspace multi-clustering methods. **a** De-coupled methods, **b** coupled methods

exact groups from the friendship network. Subspace multi-clustering methods can help find different communities in different subspaces. For example, different groups of people from the network in Fig. 5a can be discovered in Fig. 5b–d in different subspaces, that is, colleagues, classmates, and family, respectively.

1.2 Categorization

Subspace multi-clustering methods aim to discover multiple non-redundant clustering structures in different feature subspaces. Considering the procedure of generating subspaces and clusterings, two categories of approaches have been proposed by researchers.

- *De-coupled methods* As shown in Fig. 6a, de-coupled methods have two stages. In the first stage, multiple different subspaces are discovered. Then, in the second stage, a traditional clustering method can be applied to the data in each subspace to obtain the corresponding clustering. Different techniques (e.g., feature transformation, dimensionality reduction, and feature selection) have been used to find subspaces.
- *Coupled methods* Fig. 6b depicts the framework of the coupled methods. These methods usually define a specific model (e.g., density-based model, subspace-dissimilarity-based model, and generative model) on the data. The clustering model facilitates the findings of the subspaces and the corresponding clustering results simultaneously.

The details of each category will be discussed in Sects. 4 and 5. Table 1 provides a summary on the subspace multi-clustering methods.

We will also discuss specific properties for subspace multi-clustering methods and focus on the following properties. (Table 2 summarizes the properties of each subspace multi-clustering method. Detailed elaborations are provided in Sects. 4 and 5.)

Table 1 Categorizations of subspace multi-clustering methods

Categories	Major techniques	Example methods
De-coupled methods	Feature transformation	[20,24,61]
	Dimensionality reduction	[22,23,76]
	Feature selection	[15,36,37,45]
Coupled methods	Density-based model	[5,17,44,56]
	Subspace-dissimilarity-based model	[58]
	Generative model	[34]

- *Input parameters (Input Paras.)* Different methods take different types of parameters such as the number of clusters, the number of clusterings, or a reference clustering.
- *Multi-clustering generation (Gen.)* We are concerned about how multiple clusterings are generated, simultaneously or sequentially.
- *Clustering redundancy* This property is to indicate if a method explicitly or implicitly constrains the differences between multiple clusterings to reduce the redundancy.
- *Scalability (Scal.)* This property is about if a method is scalable on high-dimensional or large-scale data.
- *Subspace* In this property, we are interested in what kind of subspaces that users can use for understanding the corresponding clustering.

1.3 Organization

Several excellent reviews on multi-clustering techniques are available. Müller et al. [55] conducted a tutorial on the topic of discovering multiple clustering solutions. A recent book on data clustering [3] includes a chapter on alternative clustering analysis [10]. However, there was no survey that focused on the subject of subspace multi-clustering.

We provide a review on subspace multi-clustering methods. In the next section, we review their connections to some related work. Section 3 formulates the problem of multi-clustering and subspace multi-clustering. Sections 4 and 5 elaborate the details of de-coupled and coupled methods, respectively. Section 6 provides several open challenges and future directions, which is followed by the conclusions in Sect. 7.

2 Multi-clustering, subspace clustering, and multi-view/multi-source clustering

In this section, we examine the connections between subspace multi-clustering and three other well-related directions, namely multi-clustering in original full feature space, subspace clustering, and multi-view/multi-source clustering.

2.1 Multi-clustering in original full feature space

In general, there are two kinds of methods for multi-clustering in original full feature space. Unsupervised methods try to simultaneously generate multiple clusterings that are constrained to be different from each other. To obtain multiple clusterings in the original full feature space, the most straightforward approaches include: (1) applying a traditional cluster-

Table 2 Properties of different methods

	Input Paras.			Gen.		Clustering redundancy	Scal.		Subspace			
	# Clusters	# Clusterings	Reference clustering	Simultaneously	Sequentially		High-dimensional	Large-scale	Feature transformation	Dimensionality reduction	Projected feature selection	Weighted feature selection
[20,24,61]	✓		✓		✓	✓			✓			
[23]		✓		✓		✓	✓				✓	
[22]	✓		✓		✓	✓	✓			✓		
[45,56]				✓			✓				✓	
[15]		✓		✓		✓					✓	
[36]	✓				✓	✓	✓	✓				✓
[37]					✓	✓	✓	✓				✓
[76]				✓		✓				✓		
[5,17,44]				✓							✓	
[58]	✓	✓		✓		✓	✓			✓		
[34]	✓	✓		✓		✓	✓	✓			✓	

ing algorithm multiple times with different parameter settings, (2) running different clustering algorithms, and (3) a combination of the above two strategies [16]. However, these simple approaches may generate redundant clusterings that are overwhelming for users. Therefore, meta clustering [16] further finds groups of clusterings that are similar to each other and outputs a representative clustering from each group. Jain et. al [41] proposed an optimization model to balance the clustering quality and the dissimilarities between clusterings, and then simultaneously generate multiple clusterings.

Semi-supervised methods use one or more reference clusterings as the guidance to find an alternative clustering that is different from the reference clustering(s). For example, COALA [8] transforms linked pairs from the reference clustering to cannot-link constraints and then generates a good but dissimilar clustering. MAXIMUS [9] utilizes a programming model to find an alternative clustering that can maximize the dissimilarity between the new clustering and all reference clusterings.

More methods lying in the above two categories are discussed in a recent survey [10] focusing on the topic of alternative clustering. Although some methods in that survey are also included in this review, we focus on the subject of subspace multi-clustering. Subspace multi-clustering aims to find different structures hidden in different subspaces that cannot be handled by the methods limited in the original full feature space. Moreover, multi-clustering methods in original full feature space are often not applicable for high-dimensional data.

2.2 Subspace clustering

Subspace clustering is to discover multiple clusters each of which is hidden in a lower-dimensional subspace. Some approaches aim to assign each object to a unique cluster, where clusters may exist in different subspaces. Therefore, these methods determine only one clustering for given data. For example, PROCLUS [2] is based on the iterative processing of k-means and selects the most compact projection (subspace) based on the currently selected medoids. The projections are restricted to be the subsets of the original attributes. Later, Aggarwal and Yu [4] proposed the ORCLUS method to find arbitrarily oriented projections. DOC [60] is a Monte Carlo algorithm developed to iteratively compute projective clusters. PreDeCon [14] introduces the concept of local subspace preferences, which captures the main directions of high density. Recently, MrCC [18] adopts the multi-resolution indexing technique to extend the scalability to detect correlation clusters.

Some other subspace clustering methods (e.g., CLIQUE [5] and SUBCLU [44]) focus on finding subspaces which contain potential clusters. Moreover, each object can be assigned to different clusters in different subspaces. These methods output a set of subspaces, each containing at least one cluster. It is possible that a subspace in the result set contains only one cluster and the cluster may cover only a small portion of the whole data set. Strictly speaking, these approaches are more about discovering multiple clusters, rather than multiple clusterings in subspace multi-clustering where each subspace contains a partitioning of the whole data set. However, different subspaces output by these subspace clustering methods can describe totally different perspectives of the data, especially for those objects that are clustered differently in different subspaces. Therefore, we include this kind of subspace clustering methods in Sect. 5.1. In this review, we treat a set of disjoint clusters that are discovered in the same subspace by these subspace clustering methods as a clustering. A comprehensive survey about subspace clustering can be found in [59].

2.3 Multi-view/multi-source clustering

Multiple clusterings are also involved in multi-view/multi-source clustering [13,38], but from a totally different angle. Multi-view/multi-source clustering focuses on the techniques to establish a consensus clustering by combining multiple clusterings, each from one view/source. Specifically, each object can be described in different ways by using different sources, each way presenting a view on the object. For example, a web page can be represented by its text or by anchor text of inbound hyperlinks. Multiple clusterings can be obtained by utilizing each view separately. In multi-view/multi-source clustering, these clusterings are assumed to be consistent to some degree and are combined to establish a consensus solution.

It has been found that multi-view/multi-source clustering can generate a better clustering than using a single view merging all available features. Bickel and Scheffer [13] considered that the available attributes can be split into two independent subsets and iterated an interleaving EM method over the two views. Kailing et al. [46] presented an efficient density-based approach to cluster multi-represented data from sparse or unreliable sources. Later, some researchers studied spectral clustering or fuzzy clustering with multiple views [71,78]. Some explored the ensemble techniques on the consensus of distributed sources [42,52] or subspace clusterings [28,31]. Recently, Hua and Pei [38] studied a novel problem of mining mutual subspace clusters from multiple sources.

In summary, multi-view/multi-source clustering focuses on combining multiple similar clustering structures, while subspace multi-clustering is to generate multiple different clustering structures.

3 Preliminaries and problem formulation

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a given data set with n objects, where each object $\mathbf{x}_i \in \mathbb{R}^d$ is represented by d attributes from the feature set $F = \{f_1, f_2, \ldots, f_d\}$. A *clustering* with k groups $C = \{c_1, c_2, \ldots, c_k\}$ is a partitioning of those objects in X such that $\cup_{i=1}^{k} c_i \subseteq X$ and $c_i \cap c_j = \emptyset$ for any $1 \leq i < j \leq k$. Each c_i $(1 \leq i \leq k)$ is called a *cluster*.

3.1 Multi-clustering

Multi-clustering aims to discover multiple different clustering structures over X. Let $C_i = \{c_1^{(i)}, c_2^{(i)}, \ldots, c_{k_i}^{(i)}\}$ and $C_j = \{c_1^{(j)}, c_2^{(j)}, \ldots, c_{k_j}^{(j)}\}$ be two clusterings of X. The clustering quality of both C_i and C_j should be high and they should be different from each other.

Some quality measures have been defined for a clustering $C = \{c_1, c_2, \ldots, c_k\}$, e.g.,

– *Davies–Bouldin index* [25] prefers a clustering with low intra-cluster distances and high inter-cluster distances. It is defined as

$$\mathrm{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{\mathrm{dis}(\mu_i, \mu_j)}$$

where μ_i is the centroid of cluster c_i, $\mathrm{dis}(\cdot, \cdot)$ is a distance function (e.g., Minkowski distance [26]) used to measure the difference between two objects, and σ_i is the average distance of all objects in cluster c_i to its centroid μ_i. Therefore, the lower the value of Davies–Bouldin index, the better the quality of the clustering.

– *Dunn index* [29] aims to identify a clustering with dense and well-separated clusters too. It is defined differently using the ratio between the minimal inter-cluster distance to the maximal intra-cluster distance, that is,

$$\mathrm{DI} = \frac{\min_{1 \le i < j \le k} \Delta(c_i, c_j)}{\max_{1 \le r \le k} \Delta'(c_r)}$$

where $\Delta(c_i, c_j)$ (e.g., single linkage, complete linkage, or average linkage [40]) measures the distance between clusters c_i and c_j, and $\Delta'(c_r)$ (e.g., the maximum distance between objects in c_r or the average distance between all object pairs) calculates the intra-cluster distance of cluster c_r. A clustering with a higher Dunn index has better quality.

– *Silhouette coefficient* [65] measures how similar each object is to its own cluster compared to other clusters. Suppose object $\mathbf{x}_i$ is assigned to cluster c_j, let $a_i = \frac{1}{|c_j|} \sum_{\mathbf{x} \in c_j} \mathrm{dis}(\mathbf{x}, \mathbf{x}_i)$ be the average distance between $\mathbf{x}_i$ and all other objects in the same cluster and $b_i = \min_{c \neq c_j} (\frac{1}{|c|} \sum_{\mathbf{x} \in c} \mathrm{dis}(\mathbf{x}, \mathbf{x}_i))$ be the lowest average distance from $\mathbf{x}_i$ to any other clusters. Silhouette coefficient is defined by the average silhouette of all objects in the given data set X as

$$\mathrm{SC} = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

Therefore, a higher silhouette coefficient indicates better clustering quality.

We let $Q(\cdot) \in [0, 1]$ denote a clustering quality function, higher $Q(\cdot)$ indicating better quality. Similarly, $\mathrm{Sim}(\cdot, \cdot) \in [0, 1]$ denotes a similarity function over two clusterings, where a higher value means more similar and the value of 1 indicates that the two clusterings are the same. Possible measures for $\mathrm{Sim}(\cdot, \cdot)$ include

– *Purity* [53] calculates to which extent two clusterings, $C_i = \{c_1^{(i)}, c_2^{(i)}, \ldots, c_{k_i}^{(i)}\}$ and $C_j = \{c_1^{(j)}, c_2^{(j)}, \ldots, c_{k_j}^{(j)}\}$, share cluster members with each other. It is defined as

$$\mathrm{purity}(C_i, C_j) = \frac{1}{n} \sum_{c \in C_i} \max_{c' \in C_j} |c \cap c'|$$

The higher the purity, the more similar C_i and C_j.

– *Rand Index* [62] examines a series of $n(n-1)/2$ pair-wise decisions. A true positive (TP) decision assigns two clustered objects in C_j to the same cluster in C_i, and a true negative (TN) decision assigns two separated objects in C_j to different clusters in C_i. Then the Rand index is defined as

$$\mathrm{RI}(C_i, C_j) = \frac{\mathrm{TP} + \mathrm{TN}}{n(n-1)/2}$$

which measures the percentage of correct decisions. More similar clusterings have higher RI between each other.

– *Normalized Mutual Information* [19] is defined as

$$\mathrm{NMI}(C_i, C_j) = \frac{I(C_i; C_j)}{[H(C_i) + H(C_j)]/2}$$

where $I(C_i; C_j) = \sum_{c \in C_i} \sum_{c' \in C_j} P(c \cap c') \log \frac{P(c \cap c')}{P(c)P(c')}$ is the mutual information measuring the amount of information shared between C_i and C_j, $P(c)$, $P(c')$, and $P(c \cap c')$ are the probabilities of an object being in cluster $c \in C_i$, cluster $c' \in C_j$, and in the

intersection of c and c', respectively. Moreover, $H(C_i) = -\sum_{c \in C_i} P(c) \log P(c)$ is the entropy of clustering C_i, describing the amount of information carried by C_i. Therefore, higher NMI indicates higher similarity between two clusterings.

Some others can be found in [39].

We define those two types of multi-clustering problems as follows. Semi-supervised multi-clustering aims to discover one new clustering based on known clusterings in each step and more new clusterings can be generated sequentially, while unsupervised multi-clustering simultaneously generates multiple clusterings.

Definition 3.1 (*Semi-supervised Multi-clustering*). Given a data set X, a collection of m ($m \geq 1$) clusterings $I = \{C_1, \ldots, C_m\}$ over X, and a trade-off parameter $\tau \geq 0$, generate one new clustering C_o over X, such that $Q(C_o) - \tau \sum_{i=1}^{m} \mathrm{Sim}(C_i, C_o)$ is maximized.

Here, $Q(C_o) - \tau \sum_{i=1}^{m} \mathrm{Sim}(C_i, C_o)$ is maximized to produce a new clustering C_o which has a high quality $Q(C_o)$ and is also different from all reference clusterings, that is, low similarities $\sum_{i=1}^{m} \mathrm{Sim}(C_i, C_o)$ to all reference clusterings. The trade-off between the clustering quality and the dissimilarities is controlled by the parameter τ.

Definition 3.2 (*Unsupervised Multi-clustering*). Given a data set X and a trade-off parameter $\tau \geq 0$, generate m ($m \geq 2$) clusterings $O = \{C_1, \ldots, C_m\}$ over X, such that $\sum_{i=1}^{m} Q(C_i) - \tau \sum_{1 \leq i < j \leq m} \mathrm{Sim}(C_i, C_j)$ is maximized.

Here, $\sum_{i=1}^{m} Q(C_i) - \tau \sum_{1 \leq i < j \leq m} \mathrm{Sim}(C_i, C_j)$ is maximized to produce multiple clusterings, all of which have high qualities $\sum_{i=1}^{m} Q(C_i)$ and are also different from each other, that is, low similarities $\sum_{1 \leq i < j \leq m} \mathrm{Sim}(C_i, C_j)$ between each pair of them. The trade-off between their qualities and dissimilarities is controlled by the parameter τ.

3.2 Subspace multi-clustering

Semi-supervised or unsupervised problems defined above similarly exist in subspace multi-clustering problems. However, each clustering C in the subspace multi-clustering problem corresponds to a feature subspace S, denoted as (S, C). Different types of subspaces have been used, which define S differently as follows.

- *Feature transformation* [33]: The subspace for feature transformation is defined as $S \in \mathbb{R}^{d \times d}$, where a new set of d features is generated by linear combinations of all attributes from the original feature space. Hopefully, after transforming data X into subspace S by $S^\top X$, a hidden structure of X can be uncovered.
- *Dimensionality reduction* [66]: A subspace used to reduce dimensionality is defined as $S \in \mathbb{R}^{d \times l}$ where $l < d$. This kind of subspaces can map the data X from the original feature space to a lower-dimensional space by $S^\top X$, from where a hidden structure of X can be discovered.
- *Feature selection* [35]: The above two kinds of subspaces both generate a new set of features, while feature selection provides subspaces that are comprised by a subset of the original feature set F. The subspace S formed by feature selection can be constructed in two different ways. The projected subspace directly uses a subset of features from F, while the weighted subspace provides each feature of F a weight to show its importance. Both feature selection strategies are special cases of feature transformation. For example, in weighted feature selection, $S \in \mathbb{R}^{d \times d}$ is formed as $S_{ii} = w_i$, where w_i is the weight assigned to the ith feature in F, while all other elements in S are zero.

We discuss these special cases in feature selection, because they essentially retain the original feature information. These two types of feature selection methods are defined as follows, respectively.

- *Projected* $S \subset F$
- *Weighted* $S = \{(w_i, f_i) | f_i \in F, 0 \leq w_i \leq 1, 1 \leq i \leq d, \sum_{i=1}^{d} w_i = 1\}$

In subspace multi-clustering problems, the difference between clusterings is usually implicitly achieved by the difference between their subspaces. Given two clusterings (S_i, C_i) and (S_j, C_j) of X, different ways have been used to measure the similarity $\text{Sim}(S_i, S_j)$ between S_i and S_j. Two examples are as follows.

– *Orthogonality* [20] means two subspaces are orthogonal to each other. For example, if $S_i, S_j \in R^{d \times 1}$ are two mapping directions and $S_i^\top S_j = 0$, they are orthogonal and totally different.
– *Hilbert–Schmidt independence criterion* [64] measures the statistical dependence among two subspaces by

$$\text{HSIC}(X_{S_i}, X_{S_j}) = \frac{1}{(n-1)^2} tr(K_{S_i} H K_{S_j} H)$$

where X_S provides a new representation of data X in subspace S, $K_{S_i}, K_{S_j} \in R^{n \times n}$ are Gram matrices [67] (Gram matrix is a weight matrix defined as an inner product between vectors in a specific kernel space, e.g., Gaussian kernel space for X as $K_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))$, where σ is the band-width parameter), $H = I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ (I is an identity matrix and $\mathbf{1}_n$ is a column vector of size n with all 1's), and $tr(\cdot)$ denotes the trace of a matrix. Therefore, a lower value of HSIC means two subspaces are more different.

More specific similarity measures will be discussed in Sects. 4 and 5. We define the semi-supervised and unsupervised subspace multi-clustering problems as follows.

Definition 3.3 (*Semi-supervised Subspace Multi-clustering*). Given a data set X, a collection of m ($m \geq 1$) clusterings $I = \{(S_1, C_1), \ldots, (S_m, C_m)\}$ over X, and a trade-off parameter $\tau \geq 0$, generate a new clustering (S_o, C_o) over X, such that $Q(C_o) - \tau \sum_{i=1}^{m} \text{Sim}(S_i, S_o)$ is maximized.

Here, $Q(C_o) - \tau \sum_{i=1}^{m} \text{Sim}(S_i, S_o)$ is maximized to produce a new clustering C_o which has a high quality $Q(C_o)$ and its subspace is also different from all reference clusterings' subspaces, that is, low similarities $\sum_{i=1}^{m} \text{Sim}(S_i, S_o)$ to all reference subspaces. The trade-off between the clustering quality and the dissimilarities is controlled by the parameter τ.

Definition 3.4 (*Unsupervised Subspace Multi-clustering*). Given a data set X and a trade-off parameter $\tau \geq 0$, generate m ($m \geq 2$) clusterings $O = \{(S_1, C_1), \ldots, (S_m, C_m)\}$ over X, such that $\sum_{i=1}^{m} Q(C_i) - \tau \sum_{1 \leq i < j \leq m} \text{Sim}(S_i, S_j)$ is maximized.

Here, $\sum_{i=1}^{m} Q(C_i) - \tau \sum_{1 \leq i < j \leq m} \text{Sim}(S_i, S_j)$ is maximized to produce multiple clusterings, all of which have high qualities $\sum_{i=1}^{m} Q(C_i)$ and are also different from each other in their subspaces, that is, low similarities $\sum_{1 \leq i < j \leq m} \text{Sim}(S_i, S_j)$ between each pair of their subspaces. The trade-off between their qualities and dissimilarities is controlled by the parameter τ.

Different from the general problem of multi-clustering, subspace multi-clustering needs to generate two elements for each clustering: a subspace S that should not be the original

feature space and the corresponding clustering structure C. De-coupled methods generate the subspaces and clusterings separately, while coupled methods simultaneously produce these two elements. We discuss their methodologies in details in Sects. 4 and 5, respectively.

4 De-coupled methods

De-coupled methods are devoted to discover only different subspaces, each subspace representing a perspective of the data. Thereafter, any traditional clustering method can be applied to generate a high-quality clustering in each subspace. Apparently, the differences between multiple clusterings are implicitly constrained by the differences between subspaces.

Three major techniques, including feature transformation, dimensionality reduction, and feature selection, have been used to identify different subspaces. We summarize specific properties for de-coupled methods such as their input parameters, subspace formats, and scalability in Table 2. In the following, we review de-coupled methods with respect to each major technique, followed by a summary about their advantages and disadvantages.

4.1 Feature transformation-based methods

One straightforward way to discover different subspaces is to find an orthogonal subspace based on a given one. Some feature transformation-based methods focus on finding a transformation matrix $S \in R^{d \times d}$ that can map the data into a subspace orthogonal to the given one.

Cui et al. [20] presented two approaches to generate an orthogonal subspace based on a given clustering. One way is to project each object $\mathbf{x}_i$ from the jth cluster onto its cluster center μ_j and then project onto an orthogonal subspace to form a residue as

$$\mathbf{x}_i^{\text{new}} = \left(I - \frac{\mu_j \mu_j^\top}{\mu_j^\top \mu_j} \right) \mathbf{x}_i$$

Then any clustering method can be applied to the transformed data. The method can be further executed onto the new clustering result until a desired number of clusterings are obtained or the sum-squared error $\sum_{j=1}^{k} \sum_{\mathbf{x}_i^{\text{new}} \in c_j} \|\mathbf{x}_i^{\text{new}} - \mu_j\|^2$ is very small. The other way is to use PCA [43] and determine p ($p \leq k$) strong principle components of the cluster centers as $A = [\phi_1, \ldots, \phi_p] \in \mathbb{R}^{d \times p}$ and then calculate the orthogonal subspace by $S = I - A(A^\top A)^{-1} A^\top$ that is of size $d \times d$. Similarly, any traditional clustering method can be applied to the transformed data $S^\top X$, and multiple clusterings can be generated sequentially. The redundancy between clustering solutions is implicitly constrained by the orthogonality between subspaces. However, the orthogonality is only specified for two nearby subspaces, not between all pairs.

Davidson and Qi [24] adopted the distance metric learning [74] technique to obtain an alternative subspace based on a given clustering. The given clustering can automatically pose must-link (objects within the same cluster) and cannot-link (objects assigned to different clusters) constraints. Therefore, any distance metric learning method can be applied to these constraints to learn a distance metric $D \in \mathbb{R}^{d \times d}$, which can make the relationships given by the known clustering easily observable. Then, an orthogonal distance metric $S \in \mathbb{R}^{d \times d}$ can be computed by Singular Value Decomposition (SVD). Concretely, if $D = U \Sigma V$, $S = U \Sigma^{-1} V$. Thereafter, any traditional clustering method can be applied on the new transformed data $S^\top X$.

The above two methods focus on finding an alternative subspace totally different from the given one, but cannot specify which properties of the reference clustering should or should not be retained. Qi and Davidson [61] proposed a Kullback–Leibler divergence-based approach to discover an alternative subspace. In this work, a user can formally specify positive and negative feedback based on the given clustering. Specifically, given a clustering $C = \{c_1, c_2, \ldots, c_k\}$, they wanted to find a transformation $S \in \mathbb{R}^{d \times d}$ that minimizes KL divergence between the probability distribution of the original space $P(X)$ and that of the transformed space $P_S(X)$, that is,

$$\min_{S \in R^{d \times d}} \mathrm{KL}(P(X) \| P_S(X))$$
$$\text{s.t.} \quad \frac{1}{n} \sum_{i=1}^{n} \sum_{j, \mathbf{x}_i \notin c_j} \|\mathbf{x}_i - \mu_j\|_B \leq \beta$$

where $B = S^\top S$ and $\| \cdot \|_B$ is the Mahalanobis distance using B. Using the constraint, the distance to the old cluster center is enforced to be high after the transformation, and thus novel clusters can be discovered. Obviously, users can modify this constraint to specify which part of the given clustering should or should not be kept. For example, if a user wants to keep the cluster structure of c from the reference clustering, the constraint can be changed to $\frac{1}{n-|c|} \sum_{\mathbf{x} \notin c} \sum_{j, \mathbf{x} \notin c_j, c_j \neq c} \|\mathbf{x} - \mu_j\|_B \leq \beta$.

It can be easily observed that all these feature transformation-based methods are trying to find a full space transformation matrix that is of size $d \times d$; therefore, they are not applicable for high-dimensional data due to the high time and space complexity.

4.2 Dimension reduction-based methods

To handle high-dimensional data, some works are devoted to discover orthogonal lower-dimensional subspaces.

Dasgupta and Ng [23] discovered multiple subspaces by utilizing different eigenvectors of the Laplacian matrix. Specifically, given the similarity matrix $W \in R^{n \times n}$ that describes the similarity between each data pair of X, let $D_{ii} = \sum_j W_{ij}$ and the Laplacian matrix $L = D^{-1/2}(D - W)D^{-1/2}$ be normalized. By conducting eigendecomposition on L, the first $m + 1$ (m is an input, the desired number of clusterings) eigenvectors corresponding to the smallest $m + 1$ eigenvalues can be obtained. Then, the ith clustering can be produced by applying 2-means clustering to the objects represented by the $(i+1)$th eigenvector. Therefore, they reduced the dimension from d to only 1. However, each clustering solution is restricted to have two clusters.

Dang and Bailey [22] presented two methods in the semi-supervised scenario where there is a given (reference) clustering C. The first approach, regularized PCA (RPCA), aims to discover a transformation matrix $S \in R^{d \times l}$ ($l < d$), which can map data from the original feature space into a new lower-dimensional subspace. This new subspace, defined as follows, maximally preserves the global variance of the data and is also independent from the given clustering.

$$\arg \max_{S \in R^{d \times l}} \mathrm{var}(S^\top X) - \mathrm{HSIC}(S^\top X, C)$$

where Hilbert–Schmidt independence criterion [64] is used to constrain the difference between the new subspace S and C's subspace as introduced in Sect. 3.2 and the given clustering C itself is directly used as the new representation of X in C's subspace. The

problem can be solved by obtaining the l leading eigenvectors of $XX^\top - XHK_CHX^\top$, where K_C is the Gram matrix as introduced in Sect. 3.2. The resulting S is guaranteed to be global optimal when l is set. Thereafter, a new clustering can be generated by applying any traditional clustering method to the new data representation $S^\top X$. They also proposed a regularized graph-based method (RegGB). Given the similarity matrix W derived from the original feature space X, RegGB learns a novel set $Y = \{\mathbf{y}_1, \ldots \mathbf{y}_n\}$, where each $\mathbf{y}_i \in R^l$ ($l < d$) is a new representation of $\mathbf{x}_i$. Y, defined as follows, optimally preserves the local proximity of the objects and is independent from the given clustering C.

$$\arg\min_{\mathbf{y}} \sum_{i=1}^{n}\sum_{j=1}^{n} \|y_i - y_j\|^2 W_{ij} \quad \text{s.t. } S_C^\top \mathbf{y}^\top = 0$$

where $\mathbf{y} \in R^{1\times n}$ is a row of Y and $S_C \in R^{n\times 1}$ is a subspace encoding the given clustering C. Specifically, S_C maps data in X to an optimal direction that can maximize the between-cluster scatter and minimize the within-cluster scatter according to C. Constraint $S_C^\top \mathbf{y}^\top = 0$ requires the new data space $\mathbf{y}$ to be orthogonal to C's subspace. Then, any traditional clustering method can be applied to Y to generate a new clustering.

Recently, Ye et al. [76] adopted independent subspace analysis (ISA) [70] to find non-redundant lower-dimensional subspaces. Specifically, given a data set X, the number m of subspaces to find, and a combination of original features from F into m subsets (e.g., a combination of 2 subsets must be $\{f_1, \ldots, f_i\}$ and $\{f_{i+1}, \ldots, f_d\}$), ISA aims to find a demixing matrix $W \in R^{d\times d}$, such that $[X_{S_1}; \ldots; X_{S_m}] = WX$ and those new representations of X have minimal mutual information. The number of dimensions for each subspace is the cardinality of the corresponding feature subsets. Then, any traditional clustering method can be applied to each X_{S_i} to produce a clustering C_i, although the authors proposed a parameter-free clustering method to produce clusterings. However, the key challenge is how to provide a good combination of the original features. The authors started from the combination of d subsets such that each subset has only one feature, and then greedily merged two features into one subset in each trial. They applied ISA to each resulting combination. Finally, they chose the ISA result with the least compressing cost.

Apparently, the methods in this category try to discover a lower-dimensional subspace for each new clustering, which is different from that of the given clustering or other clusterings. This makes these methods scalable on high-dimensional data except for the method in [76] that needs to apply ISA to learn a $d \times d$ demixing matrix multiple times. However, this kind of subspaces generates new features that are not intuitively interpretable for users unless using additional techniques [23] to find the relationships between clusterings and the original features.

4.3 Feature selection-based methods

Although multiple clusterings can provide users different structures hidden in data, they may also overwhelm users since users need to understand each clustering structure. Finding relevant features sometimes is helpful, e.g., a clustering based on the salary feature may indicate groups of people, such as high income, medium income, and low income. Subspaces that are comprised by the original features from F are more desired in this respect.

Kailing et al. [45] defined the *interestingness* of subspaces, where a subspace is a subset of the original features. The quality of a subspace is based on the density notion of clusters. They called an object $\mathbf{x}$ a core object if its number of neighbors within distance ε is not smaller than a threshold $MinPts$, that is, $|\mathcal{N}_\varepsilon(\mathbf{x})| \geq MinPts$. Let $count[S] = \sum_{\mathbf{x}} |\mathcal{N}_\varepsilon(\mathbf{x})|$

denote the total number of points lying within distance ε from core objects in subspace S. The ratio between $count[S]$ and that when all objects were uniformly distributed in S is defined to be the interestingness of a subspace S, that is,

$$\frac{count[S]}{n \cdot \frac{Vol_{\varepsilon}^{dim[S]} \cdot n}{attrRange^{dim[S]}}}$$

where all features have been normalized within the same $attrRange$ and $Vol_{\varepsilon}^{dim[S]}$ computes the volume of S's ε-neighborhood. Then, they proposed the RIS algorithm. According to the monotonicity of the core-object condition that if $\mathbf{x}$ is a core object in S, it is also a core object in any subspace of S, RIS generates all subspaces that contain core objects in a bottom-up manner. Subspaces are further pruned using some criteria, e.g., if T is a subspace of S and S has higher quality, T is pruned. The surviving subspaces are ranked according to their interestingness as output. Thereafter, any traditional clustering method can be applied to produce a clustering in each subspace. Although different subspaces consist of different features, the relationships between their clusterings are unknown.

The cumulative mutual information (CMI) [15] method aims at selecting high-contrast subspaces that potentially provide high contrast between clusters. More specifically, let F_i denote a continuous random variable for the ith feature. For a subspace with features $\{f_1, \ldots, f_N\}$, $CMI(F_1, \ldots, F_N)$ is used to measure its quality, defined as

$$\text{CMI}(F_1, \ldots, F_N) = \sum_{i=2}^{N} \text{dif}\,(P(F_i),\, P(F_i|F_1, \ldots, F_{i-1}))$$

where dif means the difference between their cumulative entropies [27]. It can be easily observed that CMI is permutation variant. Therefore, a heuristic is applied to select a dimension permutation that approximates the optimal CMI value for a given subspace. To tackle the exponential $(2^d - 1)$ search space, they started with the two-dimensional subspaces. At each level, the top N (a predefined number) subspaces of high contrast are used to generate new candidates at the next level. A newly generated candidate subspace is considered only if all of its subspaces have high contrast. Then, the top m (an input) subspaces according to the CMI values are output. The redundancy between clusterings is implicitly constrained by the contrast between subspaces. Any traditional clustering method can be further applied in each chosen subspace.

Recently, Hu et al. [36,37] proposed to discover subspaces which can induce stable clusterings. A clustering is regarded stable if small distortions on the attribute values do not affect the discoverability of the clustering. Mathematically, the larger the eigengap of the corresponding Laplacian matrix, the more stable the clustering. Based on this stability notion, a novel multi-clustering method, named multiple stable clustering (MSC), was presented to obtain a certain number (determined by the algorithm) of stable clusterings. Specifically, they aim to find multiple different weight vectors $\mathbf{w} = [w_1, w_2, \ldots, w_d]$ from the simplex Δ^d to form multiple subspaces of format $S = \{(w_i, f_i) | f_i \in F, 0 \leq w_i \leq 1, 1 \leq i \leq d, \sum_{i=1}^{d} w_i = 1\}$. Within each subspace, the induced Laplacian matrix L has a sufficiently large eigengap between the kth and $(k+1)$th eigenvalues. After obtaining a certain number of weight vectors, any clustering method can be applied in each subspace. The difference between two clusterings is implicitly enforced by the clustering stability and the difference between weight vectors. In [37], they extended the problem which does not require a user to specify the number of clusters in each subspace.

Table 3 Specific advantages (pros) and disadvantages (cons) of de-coupled subspace multi-clustering methods

Methods	Pros	Cons
Cui et al. [20]	It can generate more than one alternative clustering	PCA is not appropriate when d is small. Clustering redundancy is only constrained between two nearby subspaces
Davidson and Qi [24]	Any distance metric learning method can be used	It can generate only one alternative clustering depending highly on the reference clustering
Qi and Davidson [61]	Users can specify which clusters to keep or reject from the given clustering	Same as Davidson and Qi [24]
Dasgupta and Ng [23]	It can simultaneously generate more than one clustering. Orthogonality between subspaces enforces non-redundancy between clusterings	Each clustering is restricted to have two clusters. Only the first clustering is optimal, while the rest clusterings are suboptimal
Dang and Bailey [22]	The subspace solution is globally optimum	Same as Davidson and Qi [24]
Ye et al. [76]	It can simultaneously generate more than one mutual independent subspaces	The input of ISA rely on approximate solutions. It needs to apply ISA multiple times and is computationally expensive
Kailing et al. [45]	Subspaces (subsets of features) are physically interpretable for users	The search space is 2^d that makes the proposal in-scalable for high-dimensional data. The relationships between clusterings from different subspaces are unknown
Böhm et al. [15]	Same as Kailing et al. [45]. CMI can be directly applied to continuous data	Both the selection of permutations and subspaces rely on approximate solutions. Users need to determine how many subspaces to keep in intermediate levels
Hu et al. [36,37]	Subspaces with feature importance are more helpful for users. It can heuristically determine the number of stable clusterings hidden in a data set	Different subspaces may generate redundant clusterings, although the subspaces themselves are not redundant

In summary, feature selection-based methods are more helpful for user understanding on each clustering. However, some methods focus only on finding *interesting* subspaces, but ignore the redundancy between their clusterings.

4.4 Summary: advantages and disadvantages

De-coupled methods attempt to highlight different structures of the data by finding different subspaces. Any traditional clustering method can be later applied in each subspace. Dissimilarities between clustering solutions are usually implicitly ensured by the differences between subspaces. However, the differences between subspaces may not directly imply the

differences between clusterings. Therefore, some de-coupled methods can produce redundant clusterings.

Feature transformation-based methods are applicable only in the situation that there is a reference clustering and they are usually impractical for high-dimensional data. Dimension reduction-based methods often generate a totally new set of features. It is hard for users to intuitively interpret the new subspaces, while feature selection-based methods are very helpful for users in this respect. In Table 3, we also summarize some specific advantages (pros) and disadvantages (cons) for each de-coupled subspace multi-clustering method.

5 Coupled methods

Unlike de-coupled methods that aim to gather new data representations in different subspaces, coupled subspace multi-clustering methods focus on generating multiple clusterings and their subspaces at the same time. We summarize specific properties for coupled methods such as their input parameters, subspace formats and scalability in Table 2. Different data models (e.g., density-based models, subspace-dissimilarity-based models, and generative models) have been used by coupled methods. In the following, we review coupled methods in each kind of data models and provide a summary about their advantages and disadvantages at the end of this section.

5.1 Density-based models

To discover multiple clusters hidden in different subspaces (subsets of features), a brute-force way is to conduct clustering in each subspace. Due to the exponentially large number, that is $2^d - 1$, of subspaces, some researches are devoted to search subspaces which have potentially dense clusters in them.

CLIQUE [5] is one of the first algorithms that attempt to find subspaces based on density. It divides each dimension into fixed grid cells by equal-length intervals. Dense cells that contain more objects than a threshold η are potentially interesting clusters. It is expensive to search all dense cells in all subspaces. Based on the monotonicity that if a cell O is dense in subspace S and subspace $T \subseteq S$, O is dense in T, CLIQUE conducts a bottom-up subspace search starting from subspaces with only one dimension. The search stops when no new subspaces can be found. Then, in each subspace, connected dense cells together form a cluster and multiple clusters can be discovered in each subspace.

ENCLUS [17] follows a similar procedure as CLIQUE [5] except that it uses the entropy to select subspaces. Let Δ be a cell in subspace S. The entropy of S is defined as

$$H(S) = -\sum_{\Delta} P(\Delta) \log P(\Delta)$$

where $P(\Delta)$ indicates the percentage of objects in Δ, that is, $\frac{|\Delta|}{n}$. A subspace whose entropy is below a predefined threshold is considered to have good clusters.

It can be observed that the above two grid-based methods highly depend on the size of the grid cells used. The number of grid cells determines the computational cost and the quality of the clustering results. Therefore, some methods [56,60,68,77] were proposed to enhance the quality of grid cells. For example, Nagesh et al. [56] proposed the MAFIA method. It also has a similar procedure as CLIQUE [5]. However, it generates adaptive grid cells based on the data distribution and does not require a user to specify the grid size. Specifically, MAFIA adaptively determines a minimum number of grid cells for each dimension. It firstly

divides each dimension into small and equal-length cells. Then, it computes the histogram for each cell and merges nearby cells with similar histograms into a new cell. This enhanced grid positioning method can speed up the search process compared to CLIQUE [5] and ENCLUS [17].

Instead of using dense grid cells, SUBCLU [44] uses a well-known spatial clustering method DBSCAN [30] to generate clusters for each candidate subspace. Therefore, the shortcomings of grid cells can be avoided. Due to the advantage of DBSCAN, arbitrarily shaped clusters can be discovered. However, it is highly inefficient due to repeating applying of DBSCAN in each subspace. Several techniques [6,7,49,54] were proposed to speed up the step of generating candidate subspaces, which make these density-based methods scalable on high-dimensional data.

Most methods in this category are subspace clustering methods, which aim to find multiple clusters hidden in different subspaces. Therefore, clusters formed in each subspace may not cover all data points. Unfortunately, many methods generate redundant clusters and the relationship between different subspaces is unknown, which is difficult to understand for users.

5.2 Subspace-dissimilarity-based models

To overcome the specific shortcomings of subspace clustering methods, Niu et al. [58] proposed a multiple spectral clustering (mSC) method. This method can simultaneously generate m (m is an input) subspaces and their corresponding cluster membership indicator matrices. Users need to choose the number of clusters in each subspace. Hilbert–Schmidt independence criterion [64] is used to quantify the correlation between two subspaces, which is incorporated into the spectral clustering optimization problem. Specifically, let $S_q \in R^{d\times l_q}$ ($1 \le q \le m, l_q \le d, \sum_q l_q \le d$) be the subspace transformation matrix, U_q be the relaxed clustering membership indicator matrix, K_q be the Gram matrix on $S_q^\top X$, and D_q be the corresponding degree matrix. They formulated the problem into the following optimization framework

$$\max_{U_1 \ldots U_m, S_1 \ldots S_m} \sum_q tr\left(U_q^\top D_q^{-1/2} K_q D_q^{-1/2} U_q\right) - \tau \sum_{r\neq q} \text{HSIC}\left(S_q^\top X, S_r^\top X\right)$$
$$\text{s.t.} \quad U_q^\top U_q = I,\ S_q^\top S_q = I$$

where the parameter τ is used to balance the clustering quality and the clustering redundancy. The number of clusters for each subspace S_q is specified in $U_q \in R^{n\times k_q}$. Consequently, m subspace transformation matrices and m relaxed clustering membership indicator matrices can be simultaneously obtained. Since U_q's are relaxed ones, k-means clustering should be further applied to each U_q to produce the corresponding clustering C_q. The dissimilarities between clusterings are implicitly constrained by the Hilbert–Schmidt independence criterion [64] between all pairs of subspaces.

5.3 Generative models

Generative models assume that data are generated from an unknown distribution. The goal of generative model-based methods is to estimate the parameters of the model from observations X, so as to find the underlying data distribution.

Günnemann et al. [34] assumed a generative model for given data using multiple mixture models. Each mixture describes a specific view (subspace) on the data. Suppose there are m

(m is an input) views, all objects are grouped in each of these views and each view provides a clustering. Specifically, latent continuous variables $V_{M,D} \in (0, 1)$ are used to reflect to which extent the Dth ($D \in \{1, \ldots, d\}$) dimension is relevant to the Mth ($M \in \{1, \ldots, m\}$) view. Then, the relevant dimensions of each subspace's cluster can be generated by the discrete random variables $G_{M,K,D} \in \{0, 1\}$ where $K \in \{1, \ldots, k\}$. $G_{M,K,D} = 1$ means the Dth dimension is relevant to the Kth cluster in the Mth view. The higher the relevance a dimension to a view, the more likely the dimension relevant to the view's cluster. This property can be realized by a Bernoulli process as

$$P\left(G_{M,K,D} = 1 | V_{M,D} = r\right) = r$$
$$P\left(G_{M,K,D} = 0 | V_{M,D} = r\right) = 1 - r$$

where the prior distribution of $V_{M,D}$ is selected according to a Beta distribution $Beta(\alpha, \beta)$. After generating the relevant dimensions of each cluster, the probability that an object $\mathbf{x}_i$ is assigned to the Kth cluster in the Mth view is defined as

$$P\left(Sel_{i,M} = K | \pi_{M,*}\right) = \pi_{M,K}$$

where the latent variable $Sel_{i,M} = K$ means that the ith object is assigned to the Kth cluster in the Mth view and $\sum_{K=1}^{k} \pi_{M,K} = 1$ for each view. However, an object may belong to two clusters that both are marked as relevant to a specific dimension. Therefore, the latent variable $Dom_{i,D} = M$ ($M \in \{1, \ldots, m\}$) is used to specify one of the views as dominant. Specifically, given $T_{i,D} = \{S_M | G_{M,Sel_{i,M},D} = 1\}$ the set of views that are potentially dominant for object $\mathbf{x}_i$ in the Dth dimension. The dominant view for object $\mathbf{x}_i$ and the Dth dimension is determined by random selection according to the following probability.

$$P\left(Dom_{i,D} = M | Sel_{i,*}, G_{*,*,D}\right) = \begin{cases} 1/|T_{i,D}| & \text{if} \quad S_M \in T_{i,D} \\ 0 & \text{if} \quad S_M \notin T_{i,D} \wedge T_{i,D} \neq \emptyset \\ 1/m & \text{o.w.} \end{cases}$$

Finally, the authors considered that the distribution of feature values $X_{i,D}$ follows a mixture of Beta distributions, that is,

$$X_{i,D} | Dom_{i,D}, Sel_{i,*}, G_{*,*,D}, \alpha_{*,*,D}, \beta_{*,*,D} \sim \begin{cases} Beta(\alpha_{M,K,D}, \beta_{M,K,D}) & \text{if} \quad V_{M,K,D} = 1 \\ Uni(0, 1) & \text{o.w.} \end{cases}$$

where $Dom_{i,D} = M$, $Sel_{i,M} = K$, and $Uni(0, 1)$ is the uniform distribution on $(0, 1)$. Thereafter, given observations X and prior distributions on π, $\alpha_{M,K,D}$ and $\beta_{M,K,D}$, the goal of this multi-view generative model (MVGen) is to maximize the a posterior probability $P(V, G, \alpha, \beta, Dom, Sel, \pi | X)$, which is solved by separating the objective function into two phases: (1) finding the best realization for V, Dom, Sel, and π by Bayesian model selection and (2) estimating the actual mixture components and the subspaces, G, α, and β. The clustering for each view can be obtained in Sel. The dissimilarities between clusterings are implicitly constrained by different mixtures.

5.4 Summary: advantages and disadvantages

It can be easily observed that most coupled methods are based on the data density. Clusterings produced by coupled methods are dependent on the specific data model adopted. Therefore, clusters may have specific properties according to the model, e.g., being dense. Those models that ignore the redundancies between multiple subspaces may generate redundant clusterings.

Table 4 Specific advantages (pros) and disadvantages (cons) of coupled subspace multi-clustering methods

Methods	Pros	Cons
Agrawal et al. [5] and Cheng et al. [17]	It can automatically determine both the number of subspaces and the number of clusters in each subspace^{+}	Redundant clusterings may overwhelm users*. They are not scalable on high-dimensional data due to the explosive number of dense cells and they are sensitive to the grid size and the density threshold used
Nagesh et al. [56]	Same as above $+$. Adaptive grids avoid the explosion of dense cells, and thus it is scalable on high-dimensional data	Same as above $*$. Generation of adaptive grids requires one parameter to determine whether to combine two cells and one parameter to determine the density threshold
Kailing et al. [44]	Same as above $+$. The density notion is enhanced compared to grid-based techniques. Arbitrarily shaped clusters can be discovered	Same as above $*$. Repeating using of DBSCAN makes it very inefficient
Niu et al. [58]	It can simultaneously generate multiple subspaces and clusterings$^{\#}$	Users need to determine the number clusterings needed and the number of clusters for each clustering^{-}. An additional step of k-means clustering needs to be applied for each relaxed membership indicator U_q. The dimension reduction matrix $S \in R^{d \times l_p}$ is not physically interpretable for users
Günnemann et al. [34]	Same as above #. Moreover, it is able to handle multiple views that compete against each other in overlapping dimensions	Same as above $-$

In Table 4, we summarize some specific advantages (pros) and disadvantages (cons) for each coupled method.

6 Future directions

As reviewed in this paper, subspace multi-clustering has been a productive direction. At the same time, a few directions are still open to be addressed in the future.

6.1 Determination of clustering/cluster numbers

Most subspace multi-clustering methods require a user to specify the number of clusterings needed and the number of clusters for each clustering. However, it is often hard for users to determine such parameters without substantial domain knowledge. Some measures can be used to determine the number of clusters for a clustering, e.g., Silhouette coefficient [65] and eigengap [37]. However, most strategies take high computational cost to determine the number of clusters. It is especially expensive for high-dimensional or large-scale data. Therefore, more efficient strategies are needed to determine the number of clusters for each clustering.

Some subspace multi-clustering methods [20,36,37] can heuristically determine the number of clusterings hidden in a given data set. For example, MSC [36] keeps searching different subspaces that contain stable clusterings in the simplex until no new subspace can be found. However, there is no theoretical guarantee that the number of clusterings discovered by these methods matches the ground truth. This is still an open challenge.

6.2 Scalability

To echo the demands on big data, handling large-scale and/or high-dimensional data has become a hot topic. Most subspace multi-clustering methods naturally incorporate techniques (e.g., feature selection or dimensionality reduction) for the challenge of high dimensionality. However, many of them leave space for improvement on their scalability on large-scale data. Recently, Hu et al. [37] presented a potential strategy to address the large-scale challenge. They selected a reasonable number of representatives from the whole data set. Then, multiple clusterings are generated on these representatives. Thereafter, the rest data are assigned to their nearest clusters in the corresponding subspace. However, the clustering quality is to some extent compromised. Therefore, there is a great demand on efficient and effective subspace multi-clustering approaches.

6.3 Subspace multi-clustering with multiple sources/views

Data can be collected from different sources, e.g., audio, image, and text, each providing a representation of the data. Therefore, multiple clusterings can be obtained by considering each representation separately. Many previous researches focus on the techniques to establish a single clustering by combining information obtained from different sources/views. However, different sources/views may represent totally different perspectives of the data and thus provide independent structures over the data. Moreover, it is still possible that a single source/view may be too weak to provide a sufficiently stable clustering. How to obtain multiple independent clusterings from multiple views/sources will be an interesting and meaningful future direction.

6.4 Visualization

It is challenging for users to understand multiple clusterings. Intuitively interpretable subspaces provide a perspective for users to understand each clustering. However, manually checking subspaces and the corresponding clusterings one by one requires a lot of efforts especially when many clusterings are generated. Interactive visualization tools will be helpful in this respect. To the best of our knowledge, there is no such tool that is designed for the specific scenario with multiple clusterings. This will be a future direction.

7 Conclusions

We review a specific area of multi-clustering, subspace multi-clustering. We formulate the subspace multi-clustering problem and then review its methodologies. Methods within this area focus on capturing different structures of data from different subspaces. De-coupled methods aim to discover different subspaces, upon which a traditional clustering approach can be applied to obtain the corresponding clustering. Coupled methods generate different subspaces and their clusterings at the same time. In addition to the discussions on the

advantages and disadvantages of different subspace multi-clustering methods, some specific properties (e.g., input parameters, subspace formats, and scalability) of these subspace multi-clustering methods are also investigated in this review.

Subspace multi-clustering methods have strengths in real tasks where different structures are naturally hidden in different subspaces. Moreover, some methods are especially applicable for high-dimensional data. We discuss some interesting and meaningful future directions.

References

1. Abbasi AA, Younis M (2007) A survey on clustering algorithms for wireless sensor networks. Comput Commun 30(14):2826–2841
2. Aggarwal CC, Procopiuc CM, Wolf JL, Yu PS, Park JS (1999) Fast algorithms for projected clustering. In: Proceedings of ACM SIGMOD international conference on management of data, Philadelphia, PA, pp 61–72
3. Aggarwal CC, Reddy CK (eds) (2014) Data clustering: algorithms and applications. CRC Press, Boca Raton
4. Aggarwal CC, Yu PS (2000) Finding generalized projected clusters in high dimensional spaces. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, Dallas, TX, pp 70–81
5. Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD international conference on management of data, Seattle, WA, pp 94–105
6. Assent I, Krieger R, Müller E, Seidl T (2007) DUSC: dimensionality unbiased subspace clustering. In: Proceedings of the 7th IEEE international conference on data mining, Omaha, NE, pp 409–414
7. Assent I, Krieger R, Müller E, Seidl T (2008) INSCY: indexing subspace clusters with in-process-removal of redundancy. In: Proceedings of the 8th IEEE international conference on data mining, Pisa, Italy, pp 719–724
8. Bae E, Bailey J (2006) COALA: a novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: Proceedings of the 6th IEEE international conference on data mining, Hong Kong, China, pp 53–62
9. Bae E, Bailey J, Dong G (2010) A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. Data Min Knowl Disc 21(3):427–471
10. Bailey J (2013) Alternative clustering analysis: a review. In: Data clustering: algorithms and applications, pp 535–550
11. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. J Comput Biol 6(3/4):281–297
12. Berkhin P (2006) A survey of clustering data mining techniques. In: Grouping multidimensional data—recent advances in clustering, pp 25–71
13. Bickel S, Scheffer T (2004) Multi-view clustering. In: Proceedings of the 4th IEEE international conference on data mining, Brighton, UK, pp 19–26
14. Böhm C, Kailing K, Kriegel H-P, Kröger P (2004) Density connected clustering with local subspace preferences. In: Proceedings of the 4th IEEE international conference on data mining, Brighton, UK, pp 27–34
15. Böhm K, Keller F, Müller E, Nguyen HV, Vreeken J (2013) CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In: Proceedings of the 13th SIAM international conference on data mining, Austin, TX, pp 198–206
16. Caruana R, Elhawary MF, Nguyen N, Smith C (2006) Meta clustering. In: Proceedings of the 6th IEEE international conference on data mining, Hong Kong, China, pp 107–118
17. Cheng CH, Fu AW-C, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. In: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA, pp 84–93
18. Cordeiro RLF, Traina AJM, Faloutsos C, Traina C Jr (2010) Finding clusters in subspaces of very large, multi-dimensional datasets. In: Proceedings of the 26th international conference on data engineering, Long Beach, CA, pp 625–636
19. Cover TM, Thomas JA (2012) Elements of information theory. Wiley, Hoboken
20. Cui Y, Fern XZ, Dy JG (2007) Non-redundant multi-view clustering via orthogonalization. In: Proceedings of the 7th IEEE international conference on data mining, Omaha, NE, pp 133–142

21. Dang X, Bailey J (2015) A framework to uncover multiple alternative clusterings. Mach Learn 98(1–2):7–30
22. Dang XH, Bailey J (2014) Generating multiple alternative clusterings via globally optimal subspaces. Data Min Knowl Disc 28(3):569–592
23. Dasgupta S, Ng V (2010) Mining clustering dimensions. In: Proceedings of the 27th international conference on machine learning, Haifa, Israel, pp 263–270
24. Davidson I, Qi Z (2008) Finding alternative clusterings using constraints. In: Proceedings of the 8th IEEE international conference on data mining, Pisa, Italy, pp 773–778
25. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1(2):224–227
26. Deza MM, Deza E (2009) Encyclopedia of distances. Springer, Berlin
27. Di Crescenzo A, Longobardi M (2009) On cumulative entropies. J Stat Plan Inference 139(12):4072–4087
28. Domeniconi C, Al-Razgan M (2009) Weighted cluster ensembles: methods and analysis. ACM Trans Knowl Discov Data 2(4):17
29. Dunn JC (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern 3(4):32–57
30. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd international conference on knowledge discovery and data mining, Portland, OR, pp 226–231
31. Fern XZ, Brodley CE (2003) Random projection for high dimensional data clustering: a cluster ensemble approach. In: Proceedings of the 20th international conference on machine learning, Washington, DC, pp 186–193
32. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
33. Gentle JE (2007) Matrix transformations and factorizations. Matrix algebra: theory, computations, and applications in statistics, pp 173–200
34. Günnemann S, Färber I, Seidl T (2012) Multi-view clustering using mixture models in subspace projections. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, Beijing, China, pp 132–140
35. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
36. Hu J, Qian Q, Pei J, Jin R, Zhu S (2015) Finding multiple stable clusterings. In: Proceedings of the 15th IEEE international conference on data mining, Atlantic City, NJ, pp 171–180
37. Hu J, Qian Q, Pei J, Jin R, Zhu S (2017) Finding multiple stable clusterings. Knowl Inf Syst 51(3):991–1021
38. Hua M, Pei J (2012) Clustering in applications with multiple data sources—a mutual subspace clustering approach. Neurocomputing 92:133–144
39. Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
40. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
41. Jain P, Meka R, Dhillon IS (2008) Simultaneous unsupervised learning of disparate clusterings. In: Proceedings of SIAM international conference on data mining, Atlanta, GA, pp 858–869
42. Januzaj E, Kriegel H-P, Pfeifle M (2004) Scalable density-based distributed clustering. In: Proceedings of the 8th European conference on principles and practice of knowledge discovery in databases, Pisa, Italy, pp 231–244
43. Jolliffe I (2002) Principal component analysis. Wiley, Hoboken
44. Kailing K, Kriegel H-P, Kröger P (2004) Density-connected subspace clustering for high-dimensional data. In: Proceedings of the 4th SIAM international conference on data mining, Lake Buena Vista, FL, pp 246–256
45. Kailing K, Kriegel H-P, Kröger P, Wanka S (2003) Ranking interesting subspaces for clustering high dimensional data. In: Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases, Cavtat-Dubrovnik, Croatia, pp 241–252
46. Kailing K, Kriegel H-P, Pryakhin A, Schubert M (2004) Clustering multi-represented objects with noise. In: Proceedings of the 8th Pacific-Asia conference on advances in knowledge discovery and data mining, Sydney, Australia, pp 394–403
47. Katiyar V, Chand N, Soni S (2010) Clustering algorithms for heterogeneous wireless sensor network: a survey. Int J Appl Eng Res 1(2):273
48. Kim S-Y, Jung T-S, Suh E-H, Hwang H-S (2006) Customer segmentation and strategy development based on customer lifetime value: a case study. Expert Syst Appl 31(1):101–107
49. Kriegel H-P, Kröger P, Renz M, Wurst SHR (2005) A generic framework for efficient subspace clustering of high-dimensional data. In: Proceedings of the 5th IEEE international conference on data mining, Houston, TX, pp 250–257

50. Leskovec J, Lang KJ, Mahoney M (2010) Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th international conference on World wide web, Raleigh, NC, pp 631–640
51. Lew MS, Sebe N, Djeraba C, Jain R (2006) Content-based multimedia information retrieval: state of the art and challenges. ACM Trans Multimed Comput Commun Appl 2(1):1–19
52. Long B, Yu PS, Zhang ZM (2008) A general model for multiple view unsupervised learning. In: Proceedings of SIAM international conference on data mining, Atlanta, GA, pp 822–833
53. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
54. Müller E, Assent I, Krieger R, Günnemann S, Seidl T (2009) DensEst: density estimation for data mining in high dimensional spaces. In: Proceedings of SIAM international conference on data mining, Sparks, NV, pp 175–186
55. Müller E, Günnemann S, Färber I, Seidl T (2012) Discovering multiple clustering solutions: grouping objects in different views of the data. In: Proceedings of the 28th IEEE international conference on data engineering, Washington, DC, pp 1207–1210
56. Nagesh HS, Goil S, Choudhary AN (2001) Adaptive grids for clustering massive data sets. In: Proceedings of the 1st SIAM international conference on data mining, Chicago, IL, pp 1–17
57. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582
58. Niu D, Dy JG, Jordan MI (2010) Multiple non-redundant spectral clustering views. In: Proceedings of the 27th international conference on machine learning, Haifa, Israel, pp 831–838
59. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. SIGKDD Explor 6(1):90–105
60. Procopiuc CM, Jones M, Agarwal PK, Murali TM (2002) A Monte Carlo algorithm for fast projective clustering. In: Proceedings of the 2002 ACM SIGMOD international conference on management of data, Madison, WI, pp 418–427
61. Qi Z, Davidson I (2009) A principled and flexible framework for finding alternative clusterings. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France, pp 717–726
62. Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
63. Raychaudhuri S, Sutphin PD, Chang JT, Altman RB (2001) Basic microarray analysis: grouping and feature reduction. Trends Biotechnol 19(5):189–193
64. Rényi A (1957) Representations for real numbers and their ergodic properties. Acta Math Hung 8(3–4):477–493
65. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
66. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
67. Schwerdtfeger H (1950) Introduction to linear algebra and the theory of matrices. P. Noordhoff, Groningen
68. Sequeira K, Zaki MJ (2005) SCHISM: a new approach to interesting subspace mining. Int J Bus Intell Data Min 1(2):137–160
69. Shapiro L, Stockman GC (2001) Computer vision. Prentice Hall, Upper Saddle River
70. Szabó Z, Póczos B, Lörincz A (2012) Separation theorem for independent subspace analysis and its consequences. Pattern Recogn 45(4):1782–1791
71. Wiswedel B, Höppner F, Berthold MR (2010) Learning in parallel universes. Data Min Knowl Disc 21(1):130–152
72. Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. Ann Data Sci 2(2):165–193
73. Xu R, Wunsch D II (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
74. Yang L, Jin R (2006) Distance metric learning: a comprehensive survey, Michigan State University 2
75. Yang S, Zhang L (2017) Non-redundant multiple clustering by nonnegative matrix factorization. Mach Learn 106(5):695–712
76. Ye W, Maurus S, Hubig N, Plant C (2016) Generalized independent subspace clustering. In: Proceedings of the 16th IEEE international conference on data mining, Barcelona, Spain, pp 569–578
77. Yiu ML, Mamoulis N (2003) Frequent-pattern based iterative projected clustering. In: Proceedings of the 3rd IEEE international conference on data mining, Melbourne, FL, pp 689–692
78. Zhou D, Burges CJC (2007) Spectral clustering and transductive learning with multiple views. In: Proceedings of the 24th international conference on machine learning, Corvallis, OR, pp 1159–1166

Juhua Hu is a Ph.D. candidate at the School of Computing Science of Simon Fraser University, Canada. Her research focuses on data mining on automatic information organization by feature engineering, especially unsupervised feature selection, similarity learning, multi-clustering, and time-series clustering. She received her B.Sc. and M.Sc. degrees in Computer Science from Nanjing University, Nanjing, China, in 2009 and 2012, respectively. From 2009 to 2012, she joined the LAMDA group led by Zhi-Hua Zhou, where she worked on machine learning especially semi-supervised learning, distance metric learning, and multi-instance multi-label learning.

Jian Pei is the Canada Research Chair (Tier 1) in Big Data Science and a Professor at the School of Computing Science, Simon Fraser University, Canada. He received his Ph.D. degree at the same school in 2002 under Dr. Jiawei Hans supervision. His research interests are to develop effective and efficient data analysis techniques for novel data intensive applications. He has published prolifically and is one of the most cited authors in data mining. He received a series of prestigious awards. He is also active in transferring the research outcome in his group to industry and applications. He is an editor of several esteemed journals in his areas and a passionate organizer of several premier academic conferences defining the frontiers of the areas. He is a Fellow of both ACM and IEEE.