Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 325–334

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Semi-Supervised Clustering Algorithms for Grouping Scientific Articles

Diego Vallejo-Huanga[1], Paulina Morillo[2], and Cèsar Ferri[3]

[1] Universidad Politécnica Salesiana, Department of Computer Science, Quito, Ecuador
dvallejoh@ups.edu.ec
[2] Universidad Politécnica Salesiana, Research Group IDEIAGEOCA, Quito, Ecuador
pmorillo@ups.edu.ec
[3] Universitat Politècnica de València, DSIC, València, Spain
cferri@dsic.upv.es

Abstract

Creating sessions in scientific conferences consists in grouping papers with common topics taking into account the size restrictions imposed by the conference schedule. Therefore, this problem can be considered as semi-supervised clustering of documents based on their content. This paper aims to propose modifications in traditional clustering algorithms to incorporate size constraints in each cluster. Specifically, two new algorithms are proposed to semi-supervised clustering, based on: binary integer linear programming with cannot-link constraints and a variation of the K-Medoids algorithm, respectively. The applicability of the proposed semi-supervised clustering methods is illustrated by addressing the problem of automatic configuration of conference schedules by clustering articles by similarity. We include experiments, applying the new techniques, over real conferences datasets: ICMLA-2014, AAAI-2013 and AAAI-2014. The results of these experiments show that the new methods are able to solve practical and real problems.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: Clustering with constraints, Size constraint, K-Medoids, Linear programming

1 Introduction

Machine learning is defined as a subfield of artificial intelligence (AI) that addresses the study and construction of models capable of learning from the data [22]. Unsupervised learning is a machine learning methodology whose task is to induce a function that presents hidden structure from unlabelled data. Clustering is an example task of unsupervised learning. Cluster analysis has the objective of dividing data objects into groups, so that objects within the same group are very similar to each other and different from objects in other groups [24].

In many cases the data or problems, per se, have certain implicit restrictions, which traditional clustering algorithms do not take advantage of. At present, certain restrictions of size and relations of belonging of objects to the clusters have been incorporated into the clustering process, which have

1877-0509
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.206

demonstrated that the performance of the algorithms proposed for the solution of this type of problems increases significantly [26]. In clustering with size constraints, the cluster size refers to the total number of objects in each cluster [25].

Document clustering is defined as the division of a documents collection into groups according to their content [14]. Document clustering has been applied to many fields of study, such as: information retrieval, topic detection and content tracking, all of them are intrinsically related to language [4]. For the systematic treatment of language, in this paper, natural language processing (NLP) techniques are used to characterize the documents that are intended to be grouped.

A scientific paper is a written report describing original research results and generally published in journals or scientific conferences. One of the main drawbacks that arise when organizing the sessions of a conference is the large number of topics addressed by the documents presented, which are disseminated in different areas of knowledge and structures that, a priori, seem to have no relationship. In addition, the problem becomes much more complex if the times that are allocated for each session in a conference are limited, so assigning the number of papers to be exposed in a session is restricted by a specific amount. This scenario can be categorized as a problem of document clustering with size constraints.

This work addresses the problem of the automatic generation of conference schedules by using clustering techniques oriented to the grouping of documents with size constraints. When grouping scientific documents (papers), we need to take into account similarities between some features of the papers, e.g.: abstract, title, keywords, corpus, etc. We can consider these similarities by a basic weighted averaging of the individual similarities. However, in this mixing step, we loose the property of representing the documents in an Eucludian space. Many of the existing clustering methods need to represent the instances in an Euclidean space and therefore they cannot be directly applicable for this problem. In this paper we present two new semi-supervised clustering algorithms with size constraints that are able to solve the proposed problem: CSCLP - Clustering algorithm with Size Constraints and Linear Programming, and K-MedoidsSC - K-Medoids algorithm with Size Constraints. These algorithms can group elements taking into account size constraints of the target clusters. Additionally, we only need to have a distance or dissimilarity matrix between the elements to be clustered (i.e. the algorithms do not require an Euclidean space to work).

This paper is organised as follows. Section 2 presents the previous work related to clustering algorithms with size constraints. The formalisation of the two new clustering algorithms is described in Section 3. Section 4 includes simulation results and experiments for the validation of the proposed algorithms: in the first instance on multivariate benchmarking datasets and subsequently with documentary datasets, which will represent conference papers in machine learning area. The holistic methodology proposed for document clustering is also presented in this section. Finally, concluding remarks and future work are presented in Section 5.

2 Previous work

A first approximation of clustering algorithms with size constraints is presented in [13], where the goal is to find equal sized clusters as well as clusters of different sizes, through Fuzzy C-means algorithm (K-Means variation) and Lagrange multipliers. There are other many proposes that have focused on the modification of classical partition algorithms (such as K-Means) for the incorporation of size constraints, for instance: [20] and [8]. In [10] the authors propose a constraint programming formulation of some of the most famous clustering methods: K-medoids (does not use the dissimilarity matrix as input), DBSCAN and Label Propagation.

The article [26] introduces an algorithm that takes as a starting point the K-Means or Metric Pairwise Constrained K-Means (MPCK-Means) algorithms, to transform the size constraint problem into an inte-

ger linear programming problem (ILP). Instance-level constraints are used in the form of inequalities so that this information can be incorporated into the linear programming problem. In the work [25], the authors introduce an algorithm called K-MeansS that allows clustering with size constraints for K-Means algorithm. The tests have shown that this procedure provides empirical evidence that combination of different kinds of partial information might improve the performance in constrained clustering.

Traditionally, in order to calculate the similarity between objects that have numeric attributes, these objects are represented as models that configures an Euclidean space where several distances can be applied in order to estimate similarities between elements. However, sometimes the dataset has no numeric attributes or it is necessary to mix several criteria and unify them to obtain a single metric that quantifies dissimilarities, and with this, we can derive a distance or dissimilarity matrix. In these situations we cannot directly apply clustering methods based on centroids, such as K-Means, since there is not an Euclidean space defined for the elements. One way to solve this type of problem is to use algorithms that, for the clustering process use only the dissimilarity matrix as input, such as the K-Medoids algorithm. Our new approaches only uses as inputs: the initial points as instance-level constraints and the distance/dissimilarity matrix.

3 Clustering Algorithms with Size Constraints: CSCLP and K-MedoidsSC

In this section we introduce the new semi-supervised algorithms that we will use in the experiments to solve the problem of clustering with size constraints.

3.1 CSCLP - Clustering Algorithm with Size Constraints and Linear Programming

In [26], the authors presented an algorithm that produces clusters that satisfy the initial restrictions by restructuring by means of ILP the starting clusters generated by original clustering methods (K-Means or MPCK-Means algorithm). One of the main disadvantages of this approach is the high dependence on the quality of the clusters that result from applying the clustering algorithm. In order to mitigate this disadvantage within the formulation proposed in this work, our approach uses only the initial points as pairwise constraints (as cannot-link constraints in semi-supervised clustering terminology) for the formation of the clusters, and through the use of binary integer linear programming (BILP) determine the membership and assignment of the instances to the clusters. In this way, the original clustering problem with size constraints becomes an optimisation problem, finding the solution efficiently through a new heuristic proposal. Additionally, the method of [26] is limited to problems where we can derive a Euclidean space between elements. Our proposal is more general since we use the distances/dissimilarities matrix as input parameter.

To formalise our problem, we have used the following notation: let $x_i = \{x_1, x_2, ..., x_n\}$ be a given dataset of n objects, where $x_i \in \mathbb{R}^m$ and $i = 1, 2, ..., n$. The size of a cluster c_j, obtained after a clustering process, is represented by its cardinality $|c_j|$. So to start the clustering process the user must specify the desire number of clusters k, for the generation of the initial points, and the size of the desired clusters $E_j = \{e_1, e_2, ..., e_k\}$ where $j = 1, 2, ..., k$. As mentioned above the k initial points $u_j = \{u_1, u_2, ..., u_k\}$ are also cannot-link constraints, which means that none of them can belong to the same cluster.

Clustering studies have shown that use of a non-random method for selection of initial points on K-Means algorithm (and its variations) can improve the results [6]. For this reason, the initial points for our clustering algorithms are chosen using two methods: Farthest Neighbour Technique [9] and

Buckshot algorithm [4]. The first one selects the k farthest points from the whole dataset, and this way the K-Medoids (or K-Means) convergence to the global optimum is insured. Buckshot is a hybrid technique whose main idea is to choose a small random sample of points (of size $\sqrt{kn}$), and then to apply a hierarchical clustering method to find k clusters. The centroids of this clusters are the k initial points. Once these initial k points are generated, an assignment of the remaining objects to one of the clusters must be found, so it is necessary to calculate the distances/dissimilarities matrix between all the objects in the dataset.

After the initial k points are determined, we need to fill the clusters by means of defining an optimisation problem. For that reason, we need to formalise properly as an ILP problem the clusters to form. Consider a matrix model, where A_{nxk} is a boolean matrix, whose elements depict the belonging of the documents to a particular group. Each row of the array represent an object x_i ($i = 1, 2, ..., n$; $n = number\ of\ objects$) and each column is a cluster c_j ($j = 1, 2, ..., k$; $k = number\ of\ clusters$). If an element of this boolean matrix a_{ij} takes a value equal to 1, it implies that the object i belongs to cluster j, otherwise, when a_{ij} takes a value equal to 0, it means that object i does not belong to cluster j.

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_k \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{nxk} \end{array}$$

The sum of elements of each row shows that an object can only belong to a single group: $\sum_{j=1}^{k} a_{ij} = 1; \forall i = 1, 2, ..., n$, this sum depicts the first constraints we are going to consider and we have called belonging constraints. The sum of elements of each column shows the size of each cluster: $\sum_{i=1}^{n} a_{ij} = e_j; \forall j = 1, 2, ..., k$, it is the second constraints we will consider and it depicts size constraints.

Following this reasoning and given the matrix of dissimilarity, we pose the problem as a linear programming problem, the objective function ($O.F.$) is given by the expression:

$$O.F. = min(\sum_{j=1}^{k} \sum_{i=1}^{n} d_{ij} p_{ij})$$

where d_{ij} is the distance of the object i to the initial object j and $p_{i,j}$ is an array of belonging of each object i to the cluster j.

3.2 K-MedoidsSC - K-Medoids algorithm with Size Constraints

One disadvantage and limitation of K-MeansS algorithm [25], is that during the iterative refinement process, the centroids are calculated and updated in a geometric space where the coordinates of the new centroids can be calculated, therefore this algorithm can not work with problems where instances not define a Euclidean space, i.e. problems where we only have a dissimilarity or distance matrix between elements in order to know their resemblance. Inspired by the work of Zhang et al.[25], we propose a change to this formulation by using the K-Medoids algorithm [17] in the clustering process and for restrict the sizes of the clusters.

The new formulation, is based on the minimisation of a cost function J_{KMS}. This function is the sum of another four cost functions. The first one J_{KM} takes into account the minimisation of the distance between objects labelled to be in a cluster and an object designated as the medoid of that cluster (K-Medoids criterion [17]). The other three are cost functions that penalise the size of the

clusters: when the desired cluster size is not achieved J_A, when cluster size is smaller than expected J_S and when the cluster size is larger than expected J_L. The function that results is given by the expression: $J_{KMS} = J_{KM} + \alpha J_A + \beta J_S + +\gamma J_L$ where α, β, γ are the corresponding non-negative scale parameters which represent different weights for the different penalty functions for the cluster sizes. In our case, we considered $\alpha = \beta = \gamma = 1$. J_A function aligns the desired cluster sizes with the obtained cluster sizes by the K-Medoids algorithm, we use the Jensen-Shannon divergence [7] to define it. J_S is defined as the sum of the distances between the cluster medoids u_j, which cluster sizes are smaller than expected, and the objects x_i that belongs to another cluster but are closer from their medoids: $J_S = \sum_{j=1}^{m} d(x_i, u_j)$ where p is the number of clusters with cluster size smaller than expected. On the other hand, J_L is defined as the sum of the distances between the cluster medoids, which cluster sizes are larger than expected, and the objects that belongs to the same cluster but are farther from their medoids: $J_L = \sum_{i=1}^{p} d(x_i, u_j)$ where m is the number of clusters with cluster size larger than expected. We modified the algorithm, so instead of calculating a new centroid for each iteration, the new algorithm chooses the object (medoid) that minimises J_{KMS} function.

Since the aforementioned penalties are applied directly to the distance or dissimilarity matrix (whose values are normalized between 0 and 1), a correction factor c_f must be applied to prevent problems with negative values. This correction factor has a domain between $1 < c_f \leqslant \infty$ and it is used to multiply all the elements of the matrix. K-MedoidsSC algorithm, then, starts an iterative process until it converges. If the number of imposed iterations has been reached, and if any of the clusters c_j does not satisfy the desired cluster size E_j, the algorithm returns to the original distance matrix to reassign the remaining or missing objects only over the clusters where their size is different to the desired group size.

If we compare K-MedoidsSC with respect to CSCLP we find important differences. CSCLP solves the semi-supervised clustering problem as an optimisation problem, while K-MedoidsSC starts from the clusters formed by K-Medoids algorithm and then the clusters are re-arranged trying to satisfy the size constrains. In the other hand, given the optimisation approach of CSCLP, probably for problems with a high number of elements to order this algorithm will not be very efficient. In this case of problems, K-MedoidsSC would be a better option.

4 Experiments

In this section we include some experiments in order to assess the performance of the proposed methods. We conduct two different settings of experiments. First, we test the validity of the methods over small and well-known datasets. Secondly, we analyse the performance over a document clustering scenario, i.e, we employ real datasets that contain data about papers of scientific conferences.

4.1 Validation of CSCLP and K-MedoidsSC Algorithms

Prior to the experimentation on documentary datasets, it is necessary to perform tests to evaluate the effectiveness and performance of the two new heuristic algorithms proposed. Three well-known classification datasets (Iris, Wine and Seeds) have been used from the UCI Machine Learning Repository (University of California Irvine). The datasets characteristics and their full descriptions can be found in [1]. All the datasets have three classes, and we use the class label as cluster identifier (the class distributions represent size constrains).

Table 1 shows a comparison of the different cluster sizes, if we apply to the datasets, algorithms without size constraints such as: Agglomerative Hierarchical Clustering - Farthest Point Algorithm (AHC-FPA) and K-Medoids together with the two new proposals with size constraints (CSCLP and K-MedoidsSC), in contrast to the real value of clusters size (determined through dataset-specific data).

Algorithm	Farthest Neighbour Technique			Buckshot Technique		
	Iris	Wine	Seeds	Iris	Wine	Seeds
AHC-FPA*	(50,29,71)	(10,32,136)	(8,42,160)	(50,29,71)	(10,32,136)	(8,42,160)
K-Medoids	(50,45,55)	(60,41,77)	(116,61,33)	(50,45,55)	(88,74,16)	(116,61,33)
CSCLP	(50,50,50)	(59,71,48)	(70,70,70)	(50,50,50)	(59,71,48)	(70,70,70)
K-MedoidsSC	(50,50,50)	(59,71,48)	(70,70,70)	(50,50,50)	(59,71,48)	(70,70,70)
Real Cluster Size	(50,50,50)	(59,71,48)	(70,70,70)	(50,50,50)	(59,71,48)	(70,70,70)
Initial Points IDs	[23,75,119]	[15,19,118]	[23,189,204]	[39,98,113]	[80,109,135]	[48,151,152]

* Does not use any technique to select initial points

Table 1: Resulting cluster sizes in datasets: Iris, Wine and Seeds, with algorithms: AHC-FPA, K-Medoids, CSCLP and K-MedoidsSC.

Datasets	Algorithm	Farthest Neighbour Technique				Buckshot Algorithm			
		ARI	AMI	NMI	$\boldsymbol{S}(i)$	ARI	AMI	NMI	$\boldsymbol{S}(i)$
Iris	AHC-FPA*	0.674	0.735	0.760	0.659	0.674	0.735	0.760	0.659
	K-Medoids	0.904	0.897	0.900	0.737	0.904	0.897	0.900	0.737
	CSCLP	0.886	0.861	0.862	0.721	0.886	0.861	0.862	0.733
	K-MedoidsSC	0.818	0.800	0.803	0.717	0.886	0.861	0.862	0.734
Wine	AHC-FPA*	0.059	0.137	0.186	0.728	0.059	0.137	0.186	0.728
	K-Medoids	0.347	0.363	0.373	0.758	0.208	0.196	0.221	0.757
	CSCLP	0.236	0.239	0.247	0.655	0.331	0.371	0.378	0.669
	K-MedoidsSC	0.302	0.297	0.304	0.716	0.347	0.374	0.380	0.699
Seeds	AHC-FPA*	0.223	0.286	0.379	0.659	0.223	0.286	0.379	0.659
	K-Medoids	0.264	0.305	0.330	0.606	0.264	0.305	0.330	0.606
	CSCLP	0.233	0.268	0.275	0.420	0.231	0.249	0.256	0.456
	K-MedoidsSC	0.149	0.179	0.186	0.348	0.162	0.189	0.196	0.276

* Does not use any technique to select initial points

Table 2: Clustering validation results in datasets: Iris, Wine and Seeds, with algorithms: AHC-FPA, K-Medoids, CSCLP and K-MedoidsSC.

In the case of K-Medoids, CSCLP and K-MedoidsSC, we compare two different methods of generating the starting cluster points: Buckshot and Farthest Neighbour Technique.

As expected, the results demonstrate that both, AHC-FPA and K-Medoids algorithms, fail to meet the expected value in cluster size since they do not use these restrictions. While the two new heuristic proposals meet the size constraints imposed by the user, matching perfectly with the real sizes of the clusters.

Since clustering algorithms define groups that are not known a priori, irrespective of the clustering methods, the final partition of data requires some kind of evaluation [11]. For this reason, and once the feasibility of the two new proposals in terms of compliance with cluster sizes has been demonstrated, it is important to validate clustering performance. Table 2 includes results of four validation measures: Adjusted Rand Index (ARI), Normalised Mutual Information (NMI), Adjusted Mutual Information (AMI), and Silhouette Coefficient $S(i)$.

External validation indices (ARI, AMI and NMI), compare properties of an algorithm's proposed clusters against that of known true clusters or "ground truth" [5]. If indices values are close to 1, it means that results of clustering are more closely to the "ground truth'. On the other hand, silhouette coefficient $S(i)$, is an internal validation index, it can be displayed in a graphical way, called silhouette diagram, which is based on the comparison of its tightness and separation. This silhouette shows which objects lie well within their cluster, and which ones are merely somewhere in between clusters. The entire clustering is displayed by combining the silhouettes into a single plot, allowing an appreciation of the relative quality of the clusters and an overview of the data configuration. The average silhouette width provides an evaluation of clustering validity [21]. The index's domain is $-1 \leq S(i) \leq 1$, therefore, a higher value of silhouette shows a better clustering performance.

The left side of Figure 1, shows the silhouette diagram of Iris dataset, and the right side indicates the dataset configuration in $\mathbb{R}^2$.

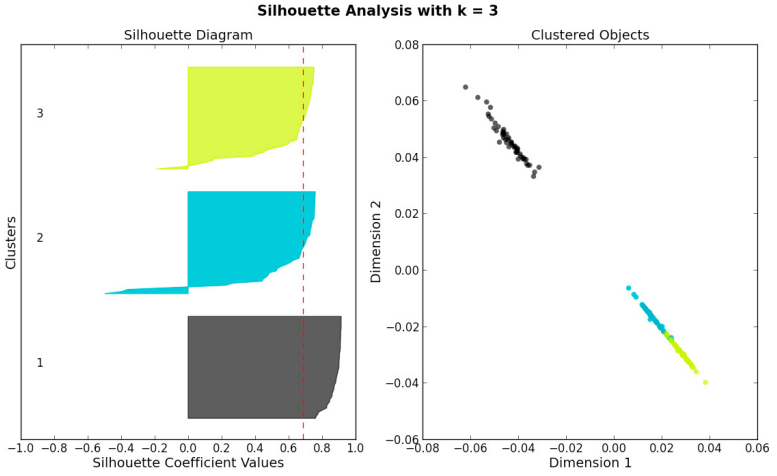

Figure 1: Clustering in Iris dataset with CSCLP algorithm (Initial points: Buckshot algorithm).

The results are notably favourable in Iris dataset, where distribution of the groups is heterogeneous and where we can clearly distinguish two groups [3] (see right plot of Figure1). But also the results in the other two datasets, Wine and Seeds, are positive according to the internal and external validation indices. It demonstrates that the algorithmic proposals presented in this paper are valid.

4.2 Experimentation in Conference Datasets

After proving that the proposed methods are able to form clusters satisfying the required size restrictions with small datasets, in this part we evaluate our methods to create conference schedules based on the similarity of the papers. Recently, a similar approach has been described by [23]. In this case the authors also use information from reviews to build the groups, and this information it is not always available.

In generic form, document clustering should be conceived as the partitioning of a documents collection into several groups according to their content [14]. Document clustering has been applied to many fields of study, such as: information retrieval, topic detection and content tracking, all of them are intrinsically related to language [4]. For the systematic treatment of language, in this paper, natural language processing (NLP) techniques are used to characterise the documents that are intended to be grouped.

A scientific article is a research paper published in specialised journals and conferences. Conferences are usually formed of various sessions where the authors present their selected papers. These sessions are usually thematic and are arranged by the conference programs chair. One of the main drawbacks that arise when organising the sessions of a conference is the large number of topics addressed by the documents presented. In addition, the problem becomes more complex if the number of papers scheduled for each session in a conference is fixed. This common scenario can be categorised as a problem of document clustering with size constraints.

In data pre-processing, NLP techniques and information retrieval models were applied to obtain a dissimilarity matrix that serves as input for the two new proposed clustering algorithms. A scientific paper is usually composed of several sections that vary according to the style of the journal, the topic area, the style of the author, etc. Machine learning articles, usually follow the Introduction, Method,

Results and Discussion (IMRaD) format, and there are three sections that have been adopted as a de facto standard: Titles, Keywords and Abstracts. According to [12], these three elements usually contribute almost 90% of the related information to the document subject and for this reason they have been used as a data source (in form of dissimilarity matrix), which characterise the differences/similarities between documents. We used a classical scheme for data pre-processing in documents: tokenization, stopwords removal and stemming [16].

Keywords are words or short-phrases (lexemes) that allow classification and orientation of indexing and retrieval systems in databases. In most scientific journals the number of keywords ranges from 3 to 10 and are usually obtained from topics-specific thesaurus [18]. On the other hand, paper's titles usually have an average of 8 to 10 words [12]. To structure the dissimilarity matrix of titles and keywords, the Jaccard coefficient has been used, since these two elements usually have a smaller number of tokens. The paper's abstract is the most consulted and read section in a scientific article [18]. This paper section usually has no more than 250 words as extension. To quantify the similarity/difference (find the dissimilarity matrix) between document abstracts, a vector model has been used with a cosine similarity index on TF-IDF weighting matrix. Thus, we have three dissimilarity matrix corresponding to titles DM_t, keywords DM_k and abstracts DM_a, and through the following equation, the criteria have been integrated to obtain a total dissimilarity matrix $DM_T = \sum_{i=1}^{3}(\theta_i)(DM_{(t,k,a)})$. θ_i is an "importance coefficient" which can be adjusted for the three dissimilarities matrix ($\sum_{i=1}^{3}\theta_i = 1$). The abstract of a paper usually provides more information than the other two elements (title and keywords) [18], so the weight of this coefficient will be greater. After several tests, we notice that the best performance is obtained with the following θ_i values: $\theta_1 = 0.55$, $\theta_2 = 0.35$ and $\theta_1 = 0.10$, for abstracts, keywords and titles, respectively. We will use these values to evaluate the performance of our algorithms.

To test the operation of the two new algorithms, in document clustering, three datasets of scientific conferences have been used. The first dataset corresponds to the "13th International Conference on Machine Learning and Applications, ICMLA-14". For the first dataset construction we used data scraping techniques from its website [15]. The other two datasets were obtained from the UCI Repository [1] corresponding to the "27th Conference on Artificial Intelligence of Association for the Advancement of Artificial Intelligence, AAAI-13" and the "28th Conference on Artificial Intelligence of Association for the Advancement of Artificial Intelligence, AAAI-14". The ICMLA-14 conference dataset consists of 69 papers, while AAAI-2013 and AAAI-2014 have 150 and 398 documents, respectively. It is important to clarify that to define the external validation indices we consider that the ground truth of classes assignment, provided by the conference schedule, is the "ideal" or "real" solution. For all the conferences, we assume that the papers have been grouped manually into sessions by the program chairs considering the similarity among papers. We examined different scenarios by combining unequal cluster sizes E_j, with different values in the number of clusters k. The performance results obtained by applying the new algorithms, CSCLP and K-MedoidsSC, over the three datasets are summarised in Table 3. Note that we do not include other semi-supervised clustering techniques in the experiments since they need a Euclidean metric defined in the elements for grouping them and here we do not have this property. The exception is the work presented in [23], but this method requires the reviews and they are not available in the studied datasets. For the size of problems analysed in this paper, in terms of execution time, both methods present a similar performance. They are able to find solutions in milliseconds.

The $S(i)$ values, for both algorithms, shown in Table 3 are appropriate, considering that the k value is high compared with the dataset size n. If we analyse the external validation indices, we see that inisialisation by Buckshot Algorithmt obtains better performance, in general, than Farthest Neighbour Technique. When comparing methods, CSCLP presents better results than K-MedoidsSC.

We also have developed a web system, called ADoCS, implementing the CSCLP algorithm that could help programs chairs to organise conference program schedules. This tool could be also useful to other related tasks in clustering documents with restrictions, such as find groups of papers to be assigned

Datasets	Algorithm	Farthest Neighbour Technique						Buckshot Algorithm					
		k	Cluster Size c_j	ARI	AMI	NMI	$S(i)$	k	Cluster Size c_j	ARI	AMI	NMI	$S(i)$
AAAI-13	CSCLP	3	(45,52,53)	0.029	0.030	0.042	0.028	3	(45,52,53)	0.036	0.037	0.049	0.031
	K-MedoidsSC	3	(45,52,53)	0.010	0.012	0.024	0.023	3	(45,52,53)	0.018	0.016	0.028	0.030
AAAI-14	CSCLP	11	(10,11,18,19,21,25,30,42,45,57,120)	0.079	0.128	0.185	0.040	11	(10,11,18,19,21,25,30,42,45,57,120)	0.074	0.120	0.178	0.037
	K-MedoidsSC	11	(10,11,18,19,21,25,30,42,45,57,120)	0.025	0.074	0.135	0.035	11	(10,11,18,19,21,25,30,42,45,57,120)	0.048	0.085	0.145	0.036
ICMLA-14	CSCLP	14	(4,5,5,5,5,5,5,5,5,5,5,5,5,5)	0.040	0.074	0.494	0.229	14	(4,5,5,5,5,5,5,5,5,5,5,5,5,5)	0.094	0.146	0.533	0.234
	K-MedoidsSC	14	(4,5,5,5,5,5,5,5,5,5,5,5,5,5)	0.063	0.107	0.512	0.230	14	(4,5,5,5,5,5,5,5,5,5,5,5,5,5)	0.048	0.079	0.497	0.217

Table 3: Clustering results in datasets: AAAI-13, AAAI-14 and ICMLA-14 with algorithms: CSCLP and K-MedoidsSC and two initial points methods (Farthest Neighbour and Buckshot).

to reviews. The web tool can be used free of charge in the url: `https://ceferra.shinyapps.io/ADoCS/`. We use an open source programming language called R [19], to implement ADoCS web tool, specifically the Shiny package [2] was used to developed the graphical user interface.

There are two ways to run the interactive web application: the first one executes the tool locally using a web browser. The second uses a free shiny application server to deploy the app on the cloud. In our case, we uploaded the application to the free server. The ADoCS source code and some datasets about conferences to test the tool can be found in `https://github.com/dievalhu/ADoCS`.

5 Conclusions

In this paper we have presented two novel algorithms for semi-supervised clustering that allow constraint the sizes of the clusters. The first one, CSCLP algorithm, is based on optimisation techniques, while the second, K-MedoidsSC algorithm, represents a variation of the original K-Medoids algorithm for considering size constrains in the clusters. Some experiments in benchmarking datasets and conference datasets have shown that the new algorithms can solve clustering problems with size constraints. We have shown the application of these methods on the automatic arranging of papers to create an appropriate conference schedule which sessions. As future work, we are interested in developing conceptual clustering methods to find topics to label the created clusters.

Acknowledgments

This work was partially supported by the the EU (FEDER) and the Spanish MINECO under grant TIN 2015-69175-C4-1-R, and by Generalitat Valenciana PROMETEOII2015/013. This work has been supported by the Secretary of Higher Education, Science and Technology (SENESCYT: Secretaría Nacional de Educación Superior, Ciencia y Tecnología), of the Republic of Ecuador.

References

[1] Catherine Blake and Christopher J. Merz. *UCI Repository of machine learning databases*. 1998. `https://archive.ics.uci.edu/ml/datasets.html`.

[2] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2016. `https://CRAN.R-project.org/package=shiny`.

[3] Marie Chavent. A monothetic clustering method. *Pattern Recognition Letters*, 19(11):989–996, 1998.

[4] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–392, 1992.

[5] Lori Dalton, Virginia Ballarin, and Marcel Brun. Clustering algorithms: on learning, validation, performance, and applications to genomics. *Current genomics*, 10(6):430–445, 2009.

[6] Usama Fayyad, Cory Reina, and Paul S. Bradley. Initialization of iterative refinement clustering algorithms. *Proceedings of ACMSIGKDD*, pages 194–198, 1998.

[7] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 31. IEEE, 2004.

[8] Nuwan Ganganath, Chi-Tsun Cheng, and Chi. K Tse. Data clustering with cluster size constraints using a modified k-means algorithm. *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), International Conference IEEE*, pages 158–161, 2014.

[9] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[10] Valerio Grossi, Anna Monreale, Mirco Nanni, Dino Pedreschi, and Franco Turini. Clustering formulation using constraint optimization. *Software Engineering and Formal Methods. Springer Berlin Heidelberg*, pages 93–107, 2015.

[11] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[12] James Hartley. New ways of making academic articles easier to read. *International Journal of Clinical and Health Psychology*, 12(1):143–160, 2012.

[13] Frank Höppner and Frank Klawonn. Clustering with size constraints. *Computational Intelligence Paradigms. Springer Berlin Heidelberg*, pages 167–180, 2008.

[14] Guobiao Hu, Shuigeng Zhou, Jihong Guan, and Xiaohua Hu. Towards effective document clustering: a constrained k-means based approach. *Information Processing & Management*, 44(4):1397–1409, 2008.

[15] ICMLA. International conference on machine learning and applications. 2014. `http://www.icmla-conference.org/icmla14/`.

[16] Monica Jha. Document clustering using k-medoids. *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, 4(1):2319–2526, 2015.

[17] Leonard Kaufman and Peter J. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1Norm and Related Methods, edited by Y. Dodge, North-Holland*, pages 405–416, 1987.

[18] Salim Mattar and Marco Gonzalez. The keys of the key words in scientific articles. *Revista MVZ Cordoba*, 17(2):2955–2956, 2011.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. `https://www.R-project.org/`.

[20] David Rebollo-Monedero, Marc Solé, Jordi Nin, and Jordi Forné. A modification of the k-means method for quasi-unsupervised learning. *Knowledge-Based Systems*, 37:176–185, 2013.

[21] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[22] Toby Segaran. *Programming collective intelligence: building smart web 2.0 applications*. O'Reilly Media, Inc., 2007.

[23] Tadej Škvorc, Nada Lavrac, and Marko Robnik-Šikonja. Co-Bidding Graphs for Constrained Paper Clustering. In *5th Symp. on Languages, Applications and Technologies (SLATE'16)*, volume 51, pages 1–13, 2016.

[24] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[25] Shaohong Zhang, Hau-San Wong, and Dongqing Xie. Semi-supervised clustering with pairwise and size constraints. *International Joint Conference on Neural Networks (IJCNN), IEEE*, pages 2450–2457, 2014.

[26] Shunzhi Zhu, Dingding Wang, and Tao. Li. Data clustering with size constraints. *Knowledge-Based Systems*, 23(8):883–889, 2010.