Knowledge-Based Systems 23 (2010) 883–889

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Data clustering with size constraints

Shunzhi Zhu [a], Dingding Wang [b], Tao Li [a,b,*]

[a] Department of Computer Science & Technology, Xiamen University of Technology, Xiamen 361024, PR China
[b] School of Computer Science, Florida International University, Miami, FL 33199, USA

ARTICLE INFO

Article history:
Received 25 January 2010
Received in revised form 29 April 2010
Accepted 13 June 2010
Available online 13 July 2010

Keywords:
Constrained clustering
Size constraints
Linear programming
Data mining
Background knowledge

ABSTRACT

Data clustering is an important and frequently used unsupervised learning method. Recent research has demonstrated that incorporating instance-level background information to traditional clustering algorithms can increase the clustering performance. In this paper, we extend traditional clustering by introducing additional prior knowledge such as the size of each cluster. We propose a heuristic algorithm to transform size constrained clustering problems into integer linear programming problems. Experiments on both synthetic and UCI datasets demonstrate that our proposed approach can utilize cluster size constraints and lead to the improvement of clustering accuracy.

1. Introduction

The goal of cluster analysis is to divide data objects into groups so that objects within a group are similar to one another and different from objects in other groups. Traditionally, clustering is viewed as an unsupervised learning method which groups data objects based only on the information presented in the dataset without any external label information [28]. *K*-means [18] is one of the simplest and most famous clustering algorithms. It defines a centroid for each cluster, which is usually the mean of a group of objects. The algorithm starts by choosing *K* initial centroids, where *K* is a user-specified number of desired clusters, and then iteratively refines and updates the centroids until there is no further change with the centroids.

In real world applications such as image coding clustering, spatial clustering in geoinformatics, and document clustering [11,14,15,20,24,26,28,17], people usually obtain some background information of the data objects' relationships or the approximate size of each group before conducting clustering. This information supposes to be very helpful in clustering the data. However, traditional clustering algorithms do not provide effective mechanisms to make use of this information.

Recent research has looked at using instance-level background information, such as pairwise must-link and cannot-link constraints. If two objects are known to be in the same group, we say that they are must-linked. Or if they are known to be in different groups, we say that they are cannot-linked. Wagstaff et al. [29,30] incorporated this type of background information to *K*-means algorithm by ensuring that constraints are satisfied at each iteration during the clustering process. Basu et al. [4,5,7,13] also considered pairwise constrains to learn an underlying metric between points while clustering. Other work on learning distance metrics for constrained clustering can be found in [6,8,9,12,25]. In addition, many methods have been developed to incorporate domain knowledge for fuzzy clustering where the data objects can be assigned to multiple clusters to various degrees (membership values). In particular, many different types of knowledge hints have been used for fuzzy clustering, including partial supervision where some data points have been labeled [23], knowledge-based indicators and guidance including proximity hints where the resemblances between some pairs of data points are provided and uncertainty hints where the confidence or difficulty of the cluster membership function for a data point is characterized [21], and domain knowledge represented in the form of a collection of viewpoints (e.g., externally introduced prototypes/representatives by users) [22]. However, little work has been reported on using the size constraints for clustering.

There is another type of work focusing on balancing constraints, i.e., clusters are of approximately the same size or importance. Besides the demands of several applications, balanced clustering is also helpful in generating more meaningful initial clusters and avoiding outlier clusters. Banerjee and Ghosh [2,3] showed that a small sample was sufficient to obtain a core clustering and then allocated the rest of the data points to the core clusters while satisfying balancing constraints. Zhong and Ghosh [32,33] also took

* Corresponding author at: School of Computer Science, Florida International University, Miami, FL 33199, USA. Tel.: +1 305 348 6036; fax: +1 305 348 3549.
E-mail addresses: szzhu@xmut.edu.cn (S. Zhu), dwang003@cs.fiu.edu (D. Wang), taoli@cs.fiu.edu (T. Li).

0950-7051/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2010.06.003

balancing constraints into consideration and developed an iterative bipartitioning heuristic for sample assignment. All of the effort has illustrated that utilizing background knowledge can improve clustering performance in accuracy and scalability.

Balancing constraints can be viewed as a special case of size constraints where all the clusters have the same size. Several real life clustering applications require the clusters that have fixed size, but not necessarily the equal size for all the clusters. For example, a typical task in marketing study is customer segmentation where customers are divided into different groups where a particular sales team or a specific amount of marketing dollars is allocated to each group. If each sales team is of different size or the allocation of marketing dollars is of different amount, then the customer segmentation problem becomes a data clustering problem with size constraints. Similarly, a job scheduling problem, where a number of jobs are assigned to different machines/processes, can be modeled as a data clustering problem with size constraints if different machines/processes have different capacities. Many other problems such as document clustering where each cluster has a fixed storage space and spatial clustering where each cluster has a specific number of spatial objects can be naturally formulated as data clustering problems with size constraints.

In this paper, we extend balancing constraints to size constraints, i.e., based on the prior knowledge of the distribution of the data, we assign the size of each cluster and try to find a partition which satisfies the size constraints. We also present some case studies of considering size constraints and instance-level cannot-link constraints simultaneously. We propose a heuristic procedure to solve these constrained clustering problems by transforming them into integer linear programming optimization problems. Experiments on synthetic and UCI datasets demonstrate the improvement of clustering accuracy using our proposed methods.

The rest of the paper is organized as follows. In Section 2, we present the problem formulation. In Section 3, we describe our heuristic procedure to produce a near-optimal solution. In Section 4, we present the experimental results. Finally, in Section 5, we conclude our work and discuss the future work.

2. Problem formulation

In the problem of clustering with size constraints, we have the prior knowledge of the number of objects in each cluster. And we can also obtain the partition result of any traditional clustering algorithm, such as *K*-means. Then the problem is formulated as follows.

Given a data set of n objects, let $A = (A_1, A_2, \ldots, A_p)$ be a known partition with p clusters, and $NumA = (na_1, na_2, \ldots, na_p)$ be the number of objects in each cluster in A. We look for another partition $B = (B_1, B_2, \ldots, B_p)$ which maximizes the agreement between A and B, and $NumB = (nb_1, nb_2, \ldots, nb_p)$ represents the size constraints, i.e., the number of objects in each cluster in B.

A and B can be represented as $n \times p$ partition matrices. Each row of the matrix represents an object, and each column is for a cluster. $a_{ij} = 1$ or $b_{ij} = 1$ when object i belongs to cluster j in partition A or B. A can be represented as

$$A = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ & & & \ldots & & & \\ & & & \ldots & & & \\ 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ & & & \ldots & & & \\ & & & \ldots & & & \\ 0 & 0 & 0 & \ldots & 0 & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where

$$\sum_{i=1}^{n} a_{ij} = na_j, \quad j = 1, \ldots, p,$$

and

$$\sum_{j=1}^{p} a_{ij} = 1, \quad i = 1, \ldots, n.$$

It is easy to see that AA^T is an $n \times n$ matrix with the values

$$(AA^T)_{ij} = \begin{cases} 1, & i \text{ and } j \text{ are in the same group in } A, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The problem is to find another partition B, which minimizes

$$\|AA^T - BB^T\|,$$

satisfying $\sum_{i=1}^{n} b_{ij} = nb_j,\ j = 1, \ldots, p$, and $\sum_{j=1}^{p} b_{ij} = 1,\ i = 1, \ldots, n$.

The problem is similar to finding a partition which maximizes its agreement with another known partition [19].

3. Solution of size constrained clustering

Now, the original size constrained clustering problem becomes an optimization problem. Here, we propose a heuristic algorithm to efficiently find the solution.

3.1. The heuristic procedure

To solve the problem stated in the previous section, first of all, we define

$$D_a = diag(na_1, na_2, \ldots, na_p), \quad (3)$$

and

$$D_b = diag(nb_1, nb_2, \ldots, nb_p). \quad (4)$$

Let

$$U_j = \frac{1}{\sqrt{na_j}} \begin{bmatrix} a_{1j} \\ a_{2j} \\ \ldots \\ \ldots \\ \ldots \\ a_{nj} \end{bmatrix}, \quad j = 1, \ldots, p, \quad (5)$$

where each $a_{ij} \in \{0, 1\}$, and

$$\sum_{i} a_{ij} = na_j, \quad j = 1, \ldots, p,$$

$$\sum_{j} a_{ij} = 1, \quad i = 1, \ldots, n.$$

Then, we can see that actually $U = A(D_a)^{-1/2}$. In the same way, let

$$V_j = \frac{1}{\sqrt{nb_j}} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \ldots \\ \ldots \\ \ldots \\ b_{nj} \end{bmatrix}, \quad j = 1, \ldots, p, \quad (6)$$

where each $b_{ij} \in \{0, 1\}$, and

$$\sum_{i} b_{ij} = nb_j, \quad j = 1, \ldots, p,$$

$$\sum_{j} b_{ij} = 1, \quad i = 1, \ldots, n.$$

Similarly, $V = B(D_b)^{-1/2}$.