Neurocomputing 142 (2014) 478–485

Contents lists available at ScienceDirect

Neurocomputing

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

Integrated constraint based clustering algorithm for high dimensional data ☆

Xinyue Liu [a], Menggang Li [b,*]

[a] School of Software, Dalian University of Technology, Dalian 116620, China
[b] China Center for Industrial Security Research, Beijing Jiaotong University, Beijing 100044, China

ARTICLE INFO

Article history:
Received 28 March 2013
Received in revised form
7 February 2014
Accepted 11 April 2014
Communicated by L. Xu
Available online 20 May 2014

Keywords:
High dimensional data
Subspace clustering
Constraint based clustering

ABSTRACT

Dimension selection, dimension weighting and data assignment are three circular dependent essential tasks for high dimensional data clustering and each such task is challenging. To meet the challenge of high dimensional data clustering, constraints have been employed in several previous works. However, these constraint based algorithms use constraints to help accomplish only one of the three essential tasks. In this paper, we propose an integrated constraint based clustering (ICBC) algorithm for high dimensional data, which exploits constraints to accomplish all the three essential tasks. Firstly we generalize the dimension selection technique of CDCDD algorithm such that dimension selection and dimension weighting could be accomplished simultaneously. Then we propose a novel constraint based data assignment method which assigns all the data points to their corresponding clusters based on the selected dimensions and dimension weights. Finally we use an optimization technique to iteratively refine the initial dimension weights and centroids, and reassign data accordingly till convergence. Experimental results on both synthetic data sets and real data sets show that our proposed ICBC algorithm outperforms typical unsupervised algorithms and other constraint based algorithms in terms of accuracy. ICBC also outperforms the other algorithms that implement dimension selection in terms of efficiency and scalability.

1. Introduction

Clustering provides a better understanding of the data by dividing data points into clusters such that objects in the same cluster are similar, whereas objects in different clusters are dissimilar, with respect to a given similarity measure. Clustering has been studied for decades and many algorithms have been proposed [19]. However, modern capabilities of automatic data generation and acquisition produce a vast amount of high dimensional data, which poses huge challenge on conventional clustering algorithms. The main challenge for clustering high dimensional data is *local feature relevance*, which means that different subsets of features are relevant for different clusters. The aim of clustering algorithms high dimensional data is to find clusters formed within their own correlated dimensions [16]. Thus high dimensional data clustering needs to solve two separated problems: the first problem is the search for the relevant subspaces and the second problem is the detection of the final clusters.

High dimensional data clustering algorithms could be categorized by their ways of dealing with local feature relevance. Subspace clustering algorithms employ a dimension selection method to form a subspace for each cluster. The performances of these algorithms are not satisfied mainly due to the following two reasons: (1) the task of selecting relevant dimensions for each cluster is very difficult, and a small error of missing relevant dimensions or including irrelevant dimensions may cause bad clustering result; (2) in these algorithms, the selected dimensions in each subspace are viewed as equally important to the corresponding cluster. However, in practical settings, the dimensions in each subspace are usually not uniformly important for the relevant cluster, thus treating the dimensions uniformly encumbers us to find clusters of real shapes. Soft subspace clustering algorithms assign different weights on all the dimensions for different clusters [13,12]. These algorithms do not assign a specific subspace and get rid of irrelevant dimensions for each cluster, thus the problem of local feature relevance is not really accounted for. The irrelevant dimensions (though usually low weighted) add noises to the procedures of finding clusters in these algorithms, leading to poor clustering results. It seems that this kind of algorithms could be adapted to include a dimension selection function by assigning

☆Supported by National Science Foundation of China (No. 6127237461300190), Program for New Century Excellent Talents in University (NCET) of China (No. NCET-11-0056), Fundamental Research Funds for the Central Universities of China (No. DUT11ZD107), Doctoral Fund of Ministry of Education of China (No. 20120041110046) and Key Project of Chinese Ministry of Education (No. 313011).
* Corresponding author. Tel.: +86 10 51688811; fax: +86 10 51683980.
E-mail addresses: xyliu_dlut@163.com (X. Liu), morganli@vip.sina.com (M. Li).

http://dx.doi.org/10.1016/j.neucom.2014.04.013
0925-2312/

some dimensions with 0 weights. Nevertheless, it is hard to determine which dimensions should be 0 weighted and till now there is no such a scheme. Moreover, there are usually a small number of relevant dimensions and very large number of irrelevant dimensions for each cluster. Thus such a scheme is inefficient. However, if we perform dimension selection firstly and then perform dimension weighting, the computation will be largely reduced. To detect the final clusters, due to the complexity of high dimensional data, most high dimensional data clustering algorithms adopt a centroid-based approach.[1] A typical centroid-based cluster detection approach firstly selects an initial centroid for each potential cluster, data points are then assigned to the closest centroid considering the relevant subspace of each centroid. The clusters are iteratively refined by updating the centroids and reassigning data points according some optimization criterion. From the above discussion, we conclude that dimension selection, dimension weighting and data assignment (initial and reassignment) are three essential tasks for high dimensional data clustering. High dimensional data clustering is challenging since each such task is hard to solve and, even worse, the three tasks are circular dependent. Some techniques implementing dimension selection, dimension weighting and data assignment jointly [22–24] have been discussed in the context of model based clustering.

In this paper, we discuss how to integrate the three tasks in one framework in the context of traditional subspace clustering with the aid of constraints. It has been pointed out that pairwise constraints in the form of must-links and cannot-links are accessible in many clustering practices [7,8,10]. There have been several work that use constraints to aid the clustering process of high dimensional data. SC-MINER [15] extends the framework of CLIQUE and ENCLUS by exploiting constraints to speed up the enumeration of subspaces. CLWC [12] incorporates constraints in the data assignment phase in a weighted k-means approach. CDCDD algorithm [25] uses constraints to find potential subspaces. These constraints based subspace clustering algorithms have shown superiorities over unsupervised (soft) subspace clustering algorithms. However, constraints are used to help accomplishing only one of the essential tasks in this algorithms. We propose an integrated constraint based high dimensional data clustering (ICBC) algorithm which uses constraint to accomplish all the three essential tasks. Firstly, we propose an integrated framework which exploits inconsistent information of constraints to select dimensions of subspace for each potential cluster and assign an initial weight on each selected dimension. This framework is generalized from CDCDD. However, CDCDD only uses constraints to accomplish the dimension selection task. Then we propose a constraint based data assignment approach through which data points are assigned to these corresponding clusters based on the selected dimensions and initial dimension weights. Note that the initial dimension weights are found with only a small amount of constraints, the clustering quality could not be very high if they are directly used to find the clusters. Thus finally, we propose an iterative optimization procedure which refines the initial dimension weights and centroids such that intra-cluster similarities are maximized, and the data points are reassigned according to the refined dimension weights and centroids. This procedure is repeated until convergence. Experimental results on both synthetic data sets and real data sets show that our proposed ICBC algorithm outperforms typical unsupervised algorithms and other constraint based algorithms in terms of accuracy. ICBC also outperforms the other algorithms that implement dimension selection in terms of efficiency and scalability.

[1] There are algorithms that are not centroid-based, e.g., density-based algorithm PreDecon [9].

2. Related work

Here we review some algorithms that are most related to our approach. One can find more comprehensive survey of high dimensional data clustering algorithms in [20,16,17].

2.1. Unsupervised algorithms

Bottom-up subspace clustering algorithms determine the subspaces that contain clusters starting from all one-dimensional subspaces, using an APRIORI style approach or other heuristics to implement a downward closure property of clusters. The pioneering algorithm is CLIQUE [3], which uses a grid-based clustering notion. Other examples of this kind of algorithms are ENCLUS [11], MAFIA [18], DUSC [4] and so on. Top-down subspace clustering algorithms determine the subspace of a cluster starting from the full-dimensional space. Some top-down algorithms assume that subspace of a cluster can be derived from the local neighborhood of the cluster center or the cluster members, e.g., PreDecon [9]. Other top-down algorithms, such as PROCLUS [1], ORCLUS [2] and FINDIT [21], use random sampling to generate a set of potential cluster centers, the clusters are refined by replacing bad cluster centers with new cluster centers as long as the clustering quality increases.

Soft subspace clustering algorithms dedicate to developing dimension weighting schemes for k-means-like approaches. LAC [13] is the first algorithm based on this idea. LAC starts with k centroids and k sets of weights. It proceeds to approximate a set of k Gaussian distributions by adapting the weights. COSA [14] does not derive a clustering but merely a similarity matrix that can be used by an arbitrary clustering algorithm afterwards. The matrix contains weights for each point specifying a subspace preference of the points. The weights for a point are determined by the average distance distribution of the k-nearest neighbors along each dimension. These approaches do not assign specific subspaces to the clusters, thus the problem of irrelevant attributes is not really accounted for.

2.2. Constraint based algorithms

In many clustering practices priori knowledge in the form of pairwise constraints (must-links and cannot-links) is accessible [7,8,10]. There have been many works that use constraints to guide the search process of traditional clustering algorithms. We refer the reader to [8] for a survey of constraint based clustering algorithms. Recently several works have integrated constraints into the clustering process for high dimensional data. SC-MINER [15] is the first such attempt. The basic idea of SC-MINER is to develop an extended bottom up framework of CLIQUE and ENCLUS by integrating instance-level constraints to speed up the enumeration of subspaces. CDCDD [25] is the first constraint based top down subspace clustering algorithm which uses the constraints information to find the feature correlation and reduce the search space. CLWC [12] is the first constraint based dimension weighting clustering algorithm. CLWC employs a weighted k-means approach very similar to LAC, but allows for incorporation of constraints in the data assignment phase.

3. Integrated constraint based clustering algorithm

3.1. Algorithm framework

We are given a data set $X=\{x_1, x_2, \ldots, x_n\}$, where the ith data point x_i is a m-dimensional vector $[x_{i1}, x_{i2}, \ldots, x_{im}]^T$. We are also given two types of constraints: must-links and cannot-links. $\mathcal{C}=\{C_1, C_2, \ldots, C_c\}$ denotes the set of constraints, $c=|\mathcal{C}|$ is the number of constraints. $ML=(x_i, x_j)$ denotes a must-link between data points x_i and x_j, $CL=\langle x_i, x_j\rangle$ denotes a cannot-link between x_i and x_j.