Expert Systems With Applications 64 (2016) 24–35

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A hash-based co-clustering algorithm for categorical data

Fabrício Olivetti de França

Center of Mathematics, Computing and Cognition (CMCC), Universidade Federal do ABC (UFABC) – Santo André, SP, Brazil

ARTICLE INFO

Article history:
Received 23 February 2015
Revised 16 July 2016
Accepted 17 July 2016
Available online 20 July 2016

Keywords:
Co-clustering
Categorical data
Data mining
Text mining
Biclustering

ABSTRACT

Cluster analysis, or clustering, refers to the analysis of the structural organization of a data set. This analysis is performed by grouping together objects of the data that are more similar among themselves than to objects of different groups. The sampled data may be described by numerical features or by a symbolic representation, known as categorical features. These features often require a transformation into numerical data in order to be properly handled by clustering algorithms. The transformation usually assigns a weight for each feature calculated by a measure of importance (i.e., frequency, mutual information). A problem with the weight assignment is that the values are calculated with respect to the whole set of objects and features. This may pose as a problem when a subset of the features have a higher degree of importance to a subset of objects but a lower degree with another subset. One way to deal with such problem is to measure the importance of each subset of features only with respect to a subset of objects. This is known as co-clustering that, similarly to clustering, is the task of finding a subset of objects and features that presents a higher similarity among themselves than to other subsets of objects and features. As one might notice, this task has a higher complexity than the traditional clustering and, if not properly dealt with, may present an scalability issue. In this paper we propose a novel co-clustering technique, called HBLCoClust, with the objective of extracting a set of co-clusters from a categorical data set, without the guarantees of an enumerative algorithm, but with the compromise of scalability. This is done by using a probabilistic clustering algorithm, named Locality Sensitive Hashing, together with the enumerative algorithm named InClose. The experimental results are competitive when applied to labeled categorical data sets and text corpora. Additionally, it is shown that the extracted co-clusters can be of practical use to expert systems such as Recommender Systems and Topic Extraction.

1. Introduction

The abundance of data being collected nowadays demands a set of tools to automatically extract useful information from them. If the data are partially labeled, a possible information to be extract is in the form of a mathematical model that can deduce the label from a set of measured variables, this characterizes the supervised learning. On the other hand, if the data are unlabeled, the information can be extracted by modeling a group structure of the objects that may describe the generating process of the data or may give a summarization of the information contained on it. This is referred as unsupervised learning and it is commonly studied by means of clustering algorithms.

Data clustering can refer to the task of dividing a data set into subset of objects that are more similar to each other than to the remaining elements of the set. There is a wide range of applications such as segmenting a surveyed population (Morgan & Sonquist, 1963) for market purposes, image quantization (Feng, Chen, & Ye, 2007), frequent patterns of gene expressions (de França & Von Zuben, 2010) and many more.

In order to accomplish such task, the objects of the data set are described by a set of features measured during the data collection. Each feature can be represented by a numerical quantity, such as height of a person, amount of gas measured on a car tank, or descriptive characteristic or category, such as the gender of a person or the research topics they are interested.

The objects described by numerical features can be conveniently represented as numerical vector and the objects can be naturally compared to each other by using distance metrics. It makes sense to say that one person has twice the height of another.

On the other hand, categorical features lacks these properties and should be transformed into numerical features in order to be compared among themselves. For example, describing one person as male and another as female does not imply that one is *more* than the other.

E-mail address: folivetti@ufabc.edu.br, fabricio.olivetti@gmail.com

http://dx.doi.org/10.1016/j.eswa.2016.07.024
0957-4174/

Some similarity metrics were proposed to quantify the difference between objects of categorical features (Boriah, Chandola, & Kumar, 2008), mostly based on the matching features between two objects. One example is the Jaccard metric that, given the sets of features for two objects, it calculates the ratio between the cardinality of the intersection between the two sets and the cardinality of their union.

One problem that must be dealt with when using categorical data is the high dimensionality. Since most clustering algorithms requires a vectorial representation of the objects, the categorical features are usually represented as a binary vector, with every position representing whether the object has a given feature or not. For example, if one measured feature is whether a person is male or female, it would be represented by a 2-dimensional vector.

But, some categorical features may span into tens, hundreds or even thousands of vector dimensions. When describing a song by its genre, each object would be represented as a vector with more than 1500 dimensions. This high dimension exponentially increases the search space and it can cause a loss of precision on the similarity metrics. This is called the Curse of Dimensionality (Har-Peled, Indyk, & Motwani, 2012).

This problem can be dealt with by reducing the dimensionality of the objects while preserving the similarity relationship between them. Probabilistic Dimension Reduction (Har-Peled et al., 2012) is the family of algorithms that exploits the probability that two similar objects will be considered to be equal when a subset of features is randomly sampled and used for comparison.

One of these algorithms, called *Minhashing* (Broder, 1997; Zamora, Mendoza, & Allende, 2016), approximates the Jaccard Index between two objects. This algorithm relies on the fact that the probability of the first non-zero position of any random permutation of the feature vector is the same for two objects is equal to the Jaccard Index between them. The algorithm generates a smaller dense representation of the data with this information.

The reduction of dimensionality has two drawbacks when applied prior the clustering procedure: i) the compact representation may hide some seemingly unimportant features that could be used to describe a smaller group and, ii) the new set of features will lose its interpretability, since there is no clear relationship between the original set and the reduced set.

A more direct approach is to perform the data clustering in a two-way manner by finding the clusters of objects conditioned to a subset of features. This defines the family of algorithms known as co-clustering.

Data Co-Clustering (Dhillon, Mallela, & Modha, 2003; de França, 2012; Gao & Akoglu, 2014; Labiod & Nadif, 2011), also known as biclustering, tries to find subsets of objects and features that maximizes the similarity of the selected objects when considering the chosen features. It exploits the fact that a given object may belong to different categories when viewed by different aspects of its description. For example, a given news text may report a story about the economies of a football team. This document will have terms that are related to sports and other terms that relates to economy. So, this object can be assigned to the group of sports related documents and the group of economy related document, depending of the selected set of words.

This technique allows for many relaxations of the constraints imposed by traditional clustering[1]. For example, each object may belong to more than one group, given a different subset of features. Additionally, one feature may be used to define different groups, associated with different subset of features.

Moreover, since each group is explicitly defined by a subset of features, the reason for grouping together a subset of objects can be easily explained, thus improving the interpretability of the model.

In many situations these relaxations can benefit the cluster analysis. When a cluster analysis is performed, it is expected that the natural grouping of the data is related to the intended labeling of the objective of the study. But, this expectation may not hold true. For example, when trying to classify a set of animals, the clustering algorithm may correctly identify that lions, deers and horses belong to the same class, but may incorrectly classify sea lions, octopus and tuna as belonging to the same class if their common traits are prevalent. The co-clustering of this data set would still find these groups but, additionally, would find other groups relating sea lions with mammals, tuna with other fishes and octopus with invertebrates.

Another possibility regards the topics extraction of a textual data set. In this task it is sought to infer the set of words that describes the topic of each document, based on the analysis of the whole data set. The usual techniques applied to such task requires a prior knowledge of how many different topics there are in the corpus and then tries to find the features responsible for the generative process of each cluster of documents. Again, the restriction of a fixed generative process for each document (i.e., the generative process of the only cluster it belongs to) may fail to acknowledge a second or third topic inside the text. With the co-clustering algorithm, the document can be grouped with different sets of documents regarding different topics and, as this procedure also highlights the features used to create each cluster, it makes the topic of each cluster explicit.

Finally, in Recommender Systems a clustering algorithm would group together users with similar tastes. But then again, only the prevalent common taste of each set of users will be taken into account. If, for example, a given user rates positively many comedy movies and just a few action movies, they would be likely grouped together with other users that share a taste for comedy. On the other hand, the co-clustering algorithm could also assign them to a group of users that like action movie. Besides, the taste of each user could be described by the combined set of features of every group they belong to.

But, this flexibility comes with a price, the number of groups that can be found is usually large, given all the possible combinations of subset of objects and features. Some of these groups may be irrelevant to the subject of analysis, rendering a burden to the post-analysis procedure.

Some co-clustering algorithms coped with this problem by reintroducing some of the constraints of the classical clustering, such as the search for a pre-specified number of clusters and assigning each object to only one group (Dhillon et al., 2003; Labiod & Nadif, 2011). These algorithms retain only the explicit description of which features were used as part of the clustering process.

Despite theses difficulties, there are some co-clustering algorithms capable of dealing with such flexibility, the most recent being the *HBLCoClust* (de França, 2012) and *CoClusLSH* (Gao & Akoglu, 2014). They both have in common the use of a probabilistic dimension reduction technique, named Locality Sensitive Hashing (LSH), used to find promising regions with high probability of containing co-clusters.

In the original *HBLCoClust* algorithm, after the search for the promising regions, a graph partitioning technique, called *METIS* (Karypis & Kumar, 2012), was used in order to generate meaningful results, but constraining the algorithm to a pre-defined number of clusters.[2]

[1] Note: these relaxations are not present in every co-clustering algorithm.

[2] It is also worth mentioning that the newest versions of *METIS* did not compile correctly in some systems, thus making the HBLCoClust inaccessible to many users.