Applied Soft Computing 41 (2016) 390–399

Contents lists available at ScienceDirect

Applied Soft Computing

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

Clustering by propagating probabilities between data points

Guojun Gan[a,*], Yuping Zhang[b], Dipak K. Dey[c]

[a] Department of Mathematics, Institute for Systems Genomics, Center for Health, Intervention, and Prevention (CHIP), University of Connecticut, 196 Auditorium Rd U-3009, Storrs, CT 06269, USA
[b] Department of Statistics, Institute for Systems Genomics, Center for Health, Intervention, and Prevention (CHIP), Center for Quantitative Medicine, University of Connecticut, 215 Glenbrook Road, U-4098, Storrs, CT 06269, USA
[c] Department of Statistics, University of Connecticut, 215 Glenbrook Road, U-4098, Storrs, CT 06269, USA

ARTICLE INFO

Article history:
Received 16 January 2015
Received in revised form
11 November 2015
Accepted 10 January 2016
Available online 28 January 2016

Keywords:
Affinity propagation
Data clustering
Graph-based clustering
Markov clustering
Probability propagation

ABSTRACT

In this paper, we propose a graph-based clustering algorithm called "probability propagation," which is able to identify clusters having spherical shapes as well as clusters having non-spherical shapes. Given a set of objects, the proposed algorithm uses local densities calculated from a kernel function and a bandwidth to initialize the probability of one object choosing another object as its attractor and then propagates the probabilities until the set of attractors become stable. Experiments on both synthetic data and real data show that the proposed method performs very well as expected.

1. Introduction

Data clustering or cluster analysis is a fundamental tool for data analysis. The goal of data clustering is to divide a set of items into groups or clusters such that items in the same cluster are more similar to each other than to items form other clusters [12,32]. As a result, data clustering has found applications in a wide range of areas such as bioinformatics [7,22], pattern recognition [20], health care [33], insurance [13,15], to just name a few.

In the past 60 years, many clustering algorithms have been developed to achieve the task of data clustering [19]. These algorithms differ significantly in terms of how clusters are defined and how the clusters are identified. The *k*-means algorithm [24] is one of the most popular and classical clustering algorithms. Used to find groups of objects with small distances among cluster members, the *k*-means algorithm starts from *k* initial cluster centers and repeats updating cluster members and cluster centers until some stopping criterion is met. The number of clusters, *k*, is a parameter of the algorithm. One drawback of the *k*-means algorithm is that it is quite sensitive to initial cluster centers, which affect clustering results and the convergence speed. For example, [26] compared four initialization methods for the *k*-means algorithm and found that random initialization is not the best method.

To address the cluster center initialization problem, Frey and Dueck [11] proposed an efficient clustering method called affinity propagation. The Affinity Propagation (AP) algorithm starts with the similarities between pairs of data points and repeats passing real-valued messages between data points until a high-quality set of exemplars (i.e., cluster centers) and corresponding clusters are found. Unlike the *k*-means algorithm [24], the AP algorithm considers simultaneously all data points as cluster centers and thus does not suffer from the cluster center initialization problem.

One drawback of the AP algorithm is that the rules of passing messages between data points are complicated. In the AP algorithm, two types of messages are exchanged between data points: the responsibility and the availability. As we will see in Section 2, the rule for updating the responsibility involves calculating the maximum of sums of the availability and the similarity; the rule for updating the availability involves calculating the sum of positive responsibilities.

Motivated by the AP algorithm, we propose in this paper a novel clustering algorithm called "probability propagation," which is able to identify clusters having spherical shapes as well as clusters having non-spherical shapes. The probability propagation (PP) algorithm starts with a matrix of probabilities calculated from local densities and keeps propagating probabilities until the set of attractors become stable. Here we use the term "attractor" to represent a cluster center because the clusters found by the PP algorithm can have non-spherical shapes.

The PP algorithm we proposed is similar to the AP algorithm and the Markov Clustering (MCL) algorithm in that all three algorithms involve certain message-passing mechanism. One major difference between the PP algorithm and the AP algorithm is that the rules of message-passing in the former are simpler than those in the later. Another difference is that the PP algorithm is able to identify clusters of non-spherical shapes but the AP algorithm cannot. One major difference between the PP algorithm and the MCL algorithm is that the PP algorithm does not use the inflation operator, which is required by the MCL algorithm. Another difference is that the stochastic matrix initialization of the PP algorithm is different from that of the MCL algorithm.

The remaining of the paper is structured as follows. In Section 2, we give a brief description of the AP algorithm, the MCL algorithm, and spectral clustering. In Section 3, we present the PP algorithm in detail. In Section 4, we demonstrate the performance of the PP algorithm by conducting experiments on both synthetic and real data sets. In Section 5, we conclude the paper and point out some areas for future research.

* Corresponding author. Tel.: +1 860 486 3919; fax: +1 860 486 4238.
E-mail addresses: Guojun.Gan@uconn.edu (G. Gan), yuping.zhang@uconn.edu (Y. Zhang), dipak.dey@uconn.edu (D.K. Dey).

http://dx.doi.org/10.1016/j.asoc.2016.01.034
1568-4946/

2. Literature review

In general, there are two types of clustering algorithms [21]: hierarchical and partitional. Hierarchical clustering algorithms produce a sequence of nested clusters organized as a hierarchical tree. Hierarchical clustering algorithms can be further classified into two types: agglomerative and divisive. An agglomerative algorithm starts from each object as a cluster and keeps merging clusters until all objects are in one cluster. In contrast, a divisive algorithm starts from all objects as one cluster and keeps splitting clusters until every cluster contains one object. Unlike hierarchical clustering algorithms, partitional clustering algorithms produce a single partition of the data instead of a sequence of partitions. The *k*-means algorithm, the AP algorithm, the MCL algorithm, and spectral clustering algorithms are partitional algorithms. In this section, we give a brief introduction to the AP algorithm, the MCL algorithm, and spectral clustering.

2.1. The AP algorithm

As we mentioned before, responsibility and availability are two types of messages exchanged between data points in the AP algorithm. The responsibility $r(i, k)$, which is sent from data point i to candidate exemplar point k, reflects how well-suited it would be for point k to be the exemplar of point i. The availability $a(i, k)$, which is sent from candidate exemplar point k to data point i, reflects how appropriate it would be for data point i to choose candidate exemplar k as its exemplar.

The rules for updating the responsibility $r(i, k)$ and the availability $a(i, k)$ are given below [11]:

$$r(i,k) \leftarrow s(i,k) - \max_{j, j\neq k}\{a(i,j) + s(i,j)\}, \tag{1}$$

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{j, j\notin\{i,k\}} \max\{0, r(j,k)\}\right\}, \quad i \neq k, \tag{2}$$

$$a(k,k) \leftarrow \sum_{j, j\neq k} \max\{0, r(j,k)\}, \tag{3}$$

where $s(i, j)$ is the similarity between points i and j for $i \neq j$ and $s(k, k)$ is an input parameter called "preference." The larger the value of $s(k, k)$, the more likely that the point k is to be chosen as an exemplar. To avoid numerical oscillations, the messages are damped according to a user-specified parameter.

In the AP algorithm, responsibilities and availabilities are updated according to the aforementioned rules repeatedly until some stop criterion is met. A simple stop criterion is to terminate the iterative process after a fixed number of iterations. At any step of the iterative process, responsibilities and availabilities can be combined to identify clusters and their members as follows. For data point i, let

$$k = \arg\max_j \{a(i,j) + r(i,j)\}.$$

Then point i is an exemplar or cluster center if $k = i$ and point k is an exemplar for point i if $k \neq i$.

2.2. The MCL algorithm

The MCL algorithm is a graph clustering algorithm developed by [30]. Given a data set, a graph is first created from the similarity matrix of the data set. The MCL algorithm starts from the stochastic matrix created from the graph and repeats manipulating the stochastic matrix until the stochastic matrix does not change.

Consider a data set with n points. Let G be a graph with n vertices corresponding to the n data points. Two vertices i and j are connected if the distance between points i and j is less than a threshold parameter δ. The graph G can be represented by an $n \times n$ matrix T_G as follows: $T_G(i, j) = 1$ if i and j are connected or 0 if otherwise. Let M be the corresponding stochastic matrix defined as:

$$M(i,j) = \frac{T_G(i,j)}{\sum_{j=1}^{n} T_G(i,j)}, \quad 1 \leq i, j \leq n.$$

The stochastic matrix is obtained by normalizing each column of the matrix T_G.

Once the stochastic matrix M is created, the MCL algorithm proceeds to update M recursively by expansion and inflation. The expansion operator Exp_t is defined as

$$\mathrm{Exp}_t M = M^t, \tag{4}$$

where t is a positive integer. The expansion operator is responsible for allowing flow to connect different regions of the graph or network. The inflation operator Γ_r is defined as

$$(\Gamma_r M)(i,j) = \frac{M^r(i,j)}{\sum_{j=1}^{n} M^r(i,j)}, \tag{5}$$

where r is a positive real number. The inflation operator raises each element of M to the rth power and then normalizes each column. The inflation operator is responsible for both strengthening and weakening current flow of information that influences the granularity of clusters. After a number of iterations, the matrix M becomes invariant under both expansion and inflation, and all non-zero elements in every column become equal.

Clusters can be formed by observing the final stochastic matrix. Let M^* be the invariant stochastic matrix obtained from the iterative process. If $M^*(i, j) = 1$, then $M^*(i, j)$ is the only non-zero entry in column j. In this case, points $\mathbf{x}_i$ and $\mathbf{x}_j$ are grouped to the same cluster. If $0 < M^*(i, j) < 1$, then there are $n/M^*(i, j)$ non-zero entries in column j. In this case, point j can be grouped with any point i with $M^*(i, j) > 0$. For real data sets, the later case usually does not happen. As long as the entries in a column of the stochastic matrix are different to each other, the inflation operator will reduce the small entries to zero.

The MCL algorithm requires two parameters t and r, which are used in Eqs. (4) and (5), respectively. The default value of t is 2. The parameter r affects the granularity of clusters. Increasing the value of r can increase the number of clusters. The default value of r is also set to 2.

The MCL algorithm can be modified to handle large data sets by pruning small entries in all columns of the stochastic matrix. In the exact implementation of the MCL algorithm, the number of operations in each iteration is $O(n^3)$, where n is the number of data points. If each column of the stochastic matrix is pruned to have at most m non-zero entries, then the number of operations in each iterations can be reduced to $O(nm^2)$. For a more detail description of the MCL algorithm, readers are referred to [29].

2.3. Spectral and Kernel clustering

Although spectral clustering algorithms are not message-passing algorithms, they are graph-based algorithms. Like the MCL algorithm, spectral clustering algorithms are able to identify clusters of arbitrary shapes [23,25,27]. Spectral clustering is also related to kernel clustering. It has been pointed out that kernel *k*-means [8] and spectral clustering are two equivalent approaches [9]. In this subsection, we present a spectral clustering algorithm. In general, a spectral clustering algorithm consists of three steps [1, Chapt. 8]: first, a similarity graph of all data points is constructed; second, the data points are mapped to a feature space in which clusters