Expert Systems With Applications 65 (2016) 304–314

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Determination of the optimal number of clusters using a spectral clustering optimization

Angel Mur[a,*], Raquel Dormido[a], Natividad Duro[a], Sebastian Dormido-Canto[a], Jesús Vega[b]

[a] *Department of Computer Sciences and Automatic Control, UNED, Juan del Rosal 16 - 28040 Madrid, Spain*
[b] *National Fusion Laboratory by Magnetic Confinement. CIEMAT, Complutense 40 - 28040 Madrid, Spain*

ARTICLE INFO

Article history:
Received 18 March 2016
Revised 23 July 2016
Accepted 20 August 2016
Available online 26 August 2016

Keywords:
Spectral clustering
Optimal number of clusters
Silhouette Index
Local scaling
Image segmentation

ABSTRACT

In this paper, we present a new method, called Spectral Global Silhouette method (*GS*), to calculate the optimal number of clusters in a dataset using a Spectral Clustering algorithm. It combines both a Silhouette Validity Index and the concept of Local Scaling. First, the *GS* algorithm has first been tested using synthetic data. Then, it is applied on real data for image segmentation task. In addition, three new methods for image segmentation and two new ways to calculate the optimal number of groups in an image are proposed. Our experiments have shown a promising performance of the proposed algorithms.

1. Introduction

Clustering is an unsupervised learning method that divides data into groups (clusters) that are meaningful and/or useful. When the objective is to find meaningful groups, then the clusters should capture the natural structure of the data (Everitt, Landau, Leese, & Stahl, 2011).

Cluster analysis plays an important role in a wide variety of fields (e.g. see Jain, Murty, & Flynn, 1999; Xu & Wunsch, 2005): social sciences, biology, statistics, pattern recognition, information retrieval, machine learning and data mining.

Several clustering methods with different characteristics have been proposed for different purposes. Some well-known clustering algorithms are: *K*means (MacQueen, 1967), *EM* (Expectation Maximization) (Dempster, Laird, & Rubin, 1977), Hierarchical clustering algorithms (*HC*) (Rokach & Maimon, 2005) and Spectral clustering (*SC*) (Luxburg, 2007; Ng, Jordan, & Weiss, 2001). In all of them, the estimation of the number of clusters contained in a dataset is an essential issue. The user has to define the number of clusters either a priori or a posteriori.

In practical problems, the number of clusters is generally unknown. A simple approach to find the optimal number consists of getting a set of data partitions with different numbers of clusters and then to select the partition that provides the best result according to a specific validity index (*VID*). With the help of this *VID* the optimal number of clusters is automatically determined.

Some of the most well-known *VID* are the Davies-Bouldin Index (Davies & Bouldin, 1979), the Calinski-Harabasz Index (Calinski & Harabasz, 1974), Dunn's Index (Dunn, 1974), the Silhouette Index (Rousseeuw, 1987), the *S_Dbw* Validity Index (Halkidi & Vazirgiannis, 2001) etc.

Spectral clustering (*SC*) is one of the most popular clustering methods. This method can be applied by using standard linear algebra techniques and it usually provides meaningful groups. It should be noted that, typically, the number of clusters is set in a manual way. However, approaches to automatically determine the optimal number of clusters are always preferred. This is one of the main objectives of the present article: determining the optimal number of clusters using a *SC* algorithm.

Ref. (Zelnik-manor & Perona, 2004) proposes a spectral clustering algorithm that computes automatically the optimal number of groups. It can also handle multi-scale data using the concept of local scaling. To determine the optimal number of groups, this algorithm minimizes the cost of aligning a set of eigenvectors with a canonical coordinate system (using rotations). Ref. (Xiang & Gong, 2008) proposes an alternative method to estimate the number of clusters. To this end, the more significant eigenvectors to get separated data (using *EM* algorithm) are selected. Both methods are applied to image segmentation.

Image segmentation is the process of assigning a label to each pixel in an image in such a way that pixels with the same label

* Corresponding author.
E-mail addresses: a.r.m.g@outlook.fr (A. Mur), raquel@dia.uned.es (R. Dormido), nduro@dia.uned.es (N. Duro), sebas@dia.uned.es (S. Dormido-Canto), jesus.vega@ciemat.es (J. Vega).

http://dx.doi.org/10.1016/j.eswa.2016.08.059
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

share certain characteristics (ex: colour, intensity, or texture). The goal of segmentation is to simplify the representation of an image into something that is meaningful and easier to analyse.

The use of Spectral clustering for image segmentation, is computationally intensive. This is due to the use of an affinity matrix (*A*) that contains all the pairwise affinities between pixels. References (Fowlkes, Belongie, Chung, & Malik, 2004; Shi & Malik, 1998; Tung, Wong, & Clausi, 2010) provide different approaches to reduce the computational requirements. For instance, (Shi & Malik, 1998) uses a sparse version of *A* in which each element is connected only to a few of its nearby neighbours in the image and all other connections are assumed to be zero. A different approach is used in (Fowlkes et al., 2004): the pairwise similarities from a small random subset of pixels are used. Last but not least, (Tung et al., 2010) combines a blockwise segmentation strategy along with a stochastic ensemble consensus.

In the present work, a simple and effective method named Spectral Global Silhouette method is shown. *GS* uses *SC* together with the Silhouette Validity index and the concept of local scaling. This combination allows finding the optimal number of clusters in a data set as well as an optimal local scaling.

The application of the *SC* algorithm to a set of data points provides new representations of these data points with the help of the largest eigenvectors of a data affinity matrix. The present article works directly on these new representations of data points unlike (Zelnik-manor & Perona, 2004) and (Xiang & Gong, 2008) that initially work with the eigenvectors either analysing their structure or selecting the more relevant.

For large datasets, the *GS* algorithm requires high computational resources. In this article, the image segmentation problem to illustrate how to apply *GS* for a large dataset is presented. Two methods are shown: *WA* and *WB*. Both of them are based on *GS* and are used to calculate the optimal number of groups in an image (*IM*). The *WA* algorithm uses a reduced version of *IM*. The *WB* algorithm uses a scalable approach. The proposed methods *WA* and *WB* are then applied to the image segmentation problem resulting in different algorithms. *WA* is applied in combination with an optimal sparse version of *A* (Shi & Malik, 1998) (*GSWA* method). Other simple method for image segmentation (*GSWB*) that uses *WB* is also presented. *GSWB* is compared and validated by means of the Nyström method (Fowlkes et al., 2004). The *WB* algorithm together with the Nyström method form the Nyström_*WB* method that is also analysed.

In Section 2 some concepts used in the paper are reviewed. In Section 3, the proposed methods *GS, WA, WB, GSWA, GSWB* and Nyström_*WB* are described. In Section 4 these methods are tested and validated using synthetic and real data. Finally in Sections 5 and 6, a discussion and conclusions of the paper are respectively presented.

2. Background

This section reviews some well-known methods used in the paper: Hierarchical clustering (*HC*) (Rokach & Maimon, 2005), *K*means clustering (MacQueen, 1967), Spectral Clustering (*SC*), Local Scaling transformation (Zelnik-manor & Perona, 2004) and the Silhouette Validity index (Rousseeuw, 1987) (Average Silhouette index (*AS*) or Simplified Silhouette index (*SS*) (Hruschka & Covões, 2005).

2.1. Hierarchical and kmeans clustering

Hierarchical clustering (*HC*) groups data over a variety of scales by creating a cluster tree or dendrogram. It follows several steps: (1) find the similarity or dissimilarity between every pair of objects in the data set (2) grouping the objects into a binary, hierarchical cluster tree (linkage) (3) determining where to cut the hierarchical tree into clusters.

In this paper, the *HC* computes the distance between two data points and the distance between two clusters (for the linkage) using respectively the Euclidean and the Average distance (this means the average of the distances of each element of the cluster with each element of the other cluster). It is important to note that this is the only choice made for the *HC* algorithm.

*K*means clustering is a partitioning method. By dividing data into *k* sub-clusters, *K*means represents all the data by the mean values or centroids of their respective sub-clusters. The selection of the initial centres in each sub-cluster is randomly chosen or derived from some heuristic. The algorithm follows an iterative process where each iteration associates every data point to its nearest centroid. This is carried out according to some chosen distance metric. The new centroids are calculated by taking the mean of all the data points within each sub-cluster. The algorithm iterates until no data points move from one sub-cluster to another.

Unlike hierarchical clustering, *K*means clustering operates on actual observations (rather than the larger set of dissimilarity measures), and creates a single level of clusters. This distinction means that *K*means clustering is often more suitable than hierarchical clustering for large amounts of data.

The algorithms *HC* and *K*means are used in the last step of the *SC* algorithm.

2.2. Spectral clustering

The goal of Spectral Clustering is to cluster a set of data points $x_1,\ldots, x_n$ as a graph partitioning problem without making any assumption on the form of the data clusters. Spectral clustering often produces better results than classical clustering algorithms such as *K*means and mixture models. It also allows finding non-convex clusters.

Different stages are involved in the Spectral Clustering algorithm. (1) A pre-processing step to construct the graph and the affinity matrix representing the data set. (2) The calculation of the spectral representation. To this end, it forms the associated Laplacian matrix and computes its eigenvalues and eigenvectors. Then, it maps each data point to a lower-dimensional representation based on two or more eigenvectors. (3) The clustering process that assign points to two or more classes, based on the new representation.

So, given a set of points $x_1,\ldots, x_n$ to be partitioned into k clusters $G_1,\ldots, G_k$ the spectral clustering can be formulated as follows (Luxburg, 2007):

1. Calculate the affinity matrix A defined by (1)

$$A(i,j) = e^{\frac{-d^2(x_i,x_j)}{2\sigma^2}} \quad for\, i \neq j$$
$$\text{and } A(i,i) = 0 \tag{1}$$

   where $d(x_i, x_j)$ is the distance between x_i and x_j and σ is a scaling parameter.
2. Construct the normalized Laplacian matrix $L_{sym} = Q^{-1/2}\ A\ Q^{-1/2}$, where Q is a diagonal matrix whose (i, i)-element is the sum of A's ith row.
3. Find the k largest eigenvectors of L_{sym} (eigenvectors whose eigenvalues are the largest in magnitude) and form the matrix U_k by stacking the eigenvectors in columns: $U_k = [u_1 \vdots \ldots \vdots u_k] \in R^{n\times k}$.
4. Form the matrix Y_k from U_k by normalizing the rows of U_k to have unit length.
5. Treat each row of Y_k as a point in R^k and cluster them into k groups $C_1,\ldots,C_k$ via *K*means (or *HC*).
6. Assign the original points x_i to cluster G_j if and only if row i of the matrix Y_k was assigned to cluster C_j.