Binary Images Clustering with K-means

João Ferreira Nunes,
Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo
Avenida do Atlântico, s/n 4900-348 Viana do Castelo, Portugal
joao.nunes@estg.ipvc.pt
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
pro09001@fc.up.pt

Abstract. Nowadays, with the large amount of image databases available, there is the need to access them in the most various and easiest ways. For many applications, an expeditious manner to fulfill such need encompasses partitioning images by their content both for indexing or retrieval purposes. In this paper we propose a method to group binary images in respect to their content by means of an unsupervised learning technique, k-means clustering. The paper describes image pre-processing and feature extraction. A clustering algorithm is then applied over the extracted feature vectors. A set of clustering quality criteria is used to assist the selection of the best number of clusters. To have a better understanding of the effectiveness of the method, the obtained clusters were evaluated against an a priori available partition of the dataset. Achieved results are encouraging and demonstrate the ability and effectiveness of the proposed approach.

Keywords: image clustering, cluster analysis, data mining, k-means.

1 Introduction

With the fast growth of multimedia data, mainly due to the wide spread of digital devices, multimedia repositories became very common, and some of them extremely large. It was natural that with this amount of archived information it would come out the need of indexing and retrieving this unstructured data. There are available some tools for managing and searching within these collections, however we also need tools to extract the hidden useful knowledge embedded on them. For example, tools for discovering relationships between objects, classifying images based on their content and tools for extracting data patterns. This paper presents some experiments on clustering binary images. We intend to demonstrate the effectiveness of the k-means clustering algorithm for grouping a set of binary silhouette images into the best number of sets, based on some features extracted from those images. Finally we also intend to evaluate the results of this method using internal and external measures.

The experiments are supported on a dataset that gives some knowledge a priori, where every image is labeled as belonging to a given set. Because these images are in a binary format and are silhouettes of objects, it is expected that after the clustering process, images from different original sets should be grouped together, and consequently the number of resulting sets it is also expected to be lower than the
initial value (e.g., silhouette images of guitar objects can be grouped together with silhouette images of spoons).

The paper is organized in four sections: the first one, which is the Introduction, is where we present the state of existing knowledge in the area of Clustering algorithms, Clustering validation and also Image Clustering. We also present the purpose of the paper, and a prediction of the expected results. The section 2 is where we can find the descriptive part, which includes the sections of Data Collection, Preprocessing and Cluster Analysis. The final results of ours experiments are reported in the third section, and finally, conclusions and future work appear in the last section.

2 Data Collection, Preprocessing and Cluster Analysis

In order to classify silhouette binary images, we have chosen the dataset “MPEG-7 Core Experiment CE-Shape-1 Test Set”. The main reason to use this image collection was the fact that it is a public dataset, and because of that, it has been used in some other studies [1] [2] [3] [4]. Another aspect that made us to decide on this dataset was the previous knowledge on the images’ labeling classification. From the beginning we knew for each image, what was its label, and that information could be useful to validate our clustering process.

With this dataset we have experimented grouping similar images into clusters through an unsupervised data mining technique of clustering [5], implemented by the k-means algorithm. However, before this step could be accomplished, previously we had to apply some image enhancements operations, so that we could work with cleaner images. In this case we have cut the images removing irrelevant information and we have also filtered morphologically them.

The workflow of our model includes the preprocessing phase, followed by the extraction of images features and finally, the clustering analysis with the k-means algorithm. The Figure 1 illustrates an overview of our clustering process.

![Image Clustering process](image)

**Figure 1.** Image Clustering process.

2.1 MPEG-7 Images Dataset

The dataset that was used in our experiments is called “MPEG-7 Core Experiment CE-Shape-1 Test Set” and was created by the MPEG-7 committee. The Motion
Picture Expert Group (MPEG) [6] is a working group of ISO/IEC and has defined the MPGE-7 standard for description and search of audio and visual content. This dataset contains a large collection of binary images, which by its definition only allocates two possible values for each pixel. In this case, the two values allocated that correspond to the image colors are the black (zero) for the background and white (one) for the foreground.

Since these silhouette images represent 2D objects that are projections of 3D objects, their silhouettes may change due to: (1) change of a view point with respect to objects; (2) non-rigid object motion (e.g., people walking or fish swimming); and (3) noise (e.g., digitization and segmentation noise). Also, few additional characteristics of the dataset to be mentioned are that some images have holes in them, while others do not, and some images have experienced a number of transformations, such as scales, cuts and rotations and, at last, the image resolution is not constant among them. Figure 2 illustrates some sample images of the MPEG-7 dataset.

Figure 2. Sample images included in “MPEG-7 Core Experiment CE-Shape-1 Test Set”. The image files with the same name prefix are classified as belonging to the same set. In this case, each image represents a different set.

The “MPEG-7 Core Experiment CE-Shape-1 Test Set” is accessible from various sources in the World Wide Web [7] [8], since it has been used on other researches, and as a result some of their authors also publish it. The dataset includes 1400 images.
grouped into 70 sets, and each set contains 20 samples. In our experiments, we wanted to work with a smaller dataset and therefore, using a visual and subjective criterion, we have chosen a MPEG-7 subset filled with the following 13 classes: bone, guitar, horse, horseshoe, heart, apple, bell, device3, cup, fork, hammer, key and spoon. This new set is composed of visually similar classes between them (guitar, spoon, key), as well as clearly distinct classes (horse, horseshoe).

### 2.2 Preprocessing and Features Extraction Phases

Preprocessing is always a necessity whenever the data to be mined is noisy, inconsistent or incomplete and preprocessing significantly improves the effectiveness of the data mining techniques [5]. This section of the paper introduces the preprocessing techniques that we have applied to the images before the feature extraction process. We intended to reduce the images’ noise by removing their irrelevant information. This was accomplished by detecting and extracting the images’ region of interest, cropping the images through their bounding box. Another preprocessing technique that we have also applied is the close morphological filter. This filter closes morphologically the binary image and it is defined as the dilation of the image followed by the erosion of the dilated image. The closing filter operation smooth boundaries, reduce small inward bumps, join narrow breaks and fill small holes caused by noise.

In order to compute these two preprocessing techniques, a procedure in MATLAB was developed and it is listed bellow:

```matlab
for i = 3 : nfiles
    % opens the image file
    img = imread(filename);
    % applys the morphological filter to the binary image
    se = strel('disk', 4);
    img = imclose(img, se);
    % REGIONPROPS expects a label matrix.
    LabeledBWImg = bwlabel(img);
    % extracts img features available from REGIONPROPS.
    stats = regionprops(LabeledBWImg);
    % crops the image through its bounding box.
    img = imcrop(img, stats.BoundingBox);
    % saves the resulting new image
    imwrite(img, filename, 'png', 'bitdepth', 1);
end
```

After accomplished the first step on the Image Clustering Process, we started to conclude which image features we intended to extract. Our main concern at this stage was to assure that all the features’ values should be normalized between zero and one, so that at the time of clustering the images, all the features would be weight balanced. Therefore we developed another MATLAB procedure that computes, for each image in the dataset, the required features based on the images attributes. Those attributes
were acquired invoking the `regionprops` function from the MATLAB Image Processing toolbox and the `momentsupto3` function from the Lifting Scheme on Quincunx Grids (LISQ) toolbox [9]. The first function measures a set of properties of the image, such as its area, Euler number, bounding box, perimeter, centroid, etc., while the second one computes the images moment invariants. The resulting features were stored on an \( n \times f \) matrix, where \( n \) is the number of images and \( f \) is the number of features and this matrix was used as input during the clustering phase. In our experiments we decided to extract seven features, which are:

F1: **Solidity.** This feature results from the ratio between the image Area and its ConvexHullArea, where Area is the number of pixels in the foreground region and the ConvexHullArea is the number of pixels of the area of the smallest convex polygon that can contain the same region.

F2: **Axis Ratio.** It’s the ratio between the MinorAxisLength and the MajorAxisLength. The MinorAxisLength gives the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region, while the MajorAxisLength gives the length (also in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region.

F3: **Areas Ratio.** It’s the ratio between the image Area and the image FilledArea. The attribute Area gives the number of pixels in its foreground region while the FilledArea gives the number of on pixels in FilledImage. This ratio feature gives a notion if the image has holes on it or not, where values close to zero indicate that the image has very few holes.

F4: **Perimeter-Area Ratio;** It’s the ratio between the image Perimeter and the image Area.

F5: **Eccentricity;** Specifies the eccentricity of the ellipse that has the same second-moments as the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. An ellipse whose eccentricity is zero is actually a circle, while an ellipse whose eccentricity is one is a line segment.

F6: **Extent;** Specifies the ratio of pixels in the foreground region with the pixels in the total bounding box. It is computed as the Area divided by the Bounding Box area.

F7: **Invariant moment;** This is a useful measure to describe objects because image properties that are found via image moments are invariant under translation, changes in scale, and also rotation. From the Hu [10] set of invariant moments, we’ve chosen the skew invariant and to compute this feature we used the `momentsupto3` function from the LISQ toolbox. This function returns all the Hu seven moments, however, we chose the one that statistically seemed to establish more differences between classes, and consequently highlight their dissimilarities.

### 2.3 Cluster Analysis

Cluster analysis is traditionally considered as an unsupervised method that has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing. The main potential of clustering is to detect the underlying structure in data, not only for classification and pattern recognition, but for model reduction and optimization. It consists in the process of grouping a set of objects into classes of similar objects [5]. Thus a cluster is a collection of data objects
that are similar to one another within the same cluster and are dissimilar to the objects
in other clusters. Similarity and dissimilarity can be measured for two objects based
on several features variables. In the context of our experiments, similarity is a
quantity that reflects the strength of the relationship between two images, and
dissimilarity measures the discrepancy between two images. Euclidean Distance
(Eq.1) is the distance measure that we’ve used to measure dissimilarities between two
images $i$ and $j$. It examines the root of square differences between all the attributes $f$
of the pair of objects (images):

$$d_{ij} = \sqrt{\sum_{f=1}^{n} (x_{ij} - x_{jj})^2}.$$  

(1)

In order to group closest images into the same sets through clustering, we’ve
implemented the k-means algorithm, which will be explained in detail in the
following section.

**K-means Algorithm**

K-means is one of the simplest unsupervised learning algorithms that solves the
clustering problem. It was developed by J. MacQueen [11] and then by J. A. Hartigan
and M. A. Wong. This algorithm is used to group objects into $k$ number of classes
based on a set of their attributes/features. It is sensitive to initialized partition. The
main idea of how this algorithm works is the following: it starts by randomly picking
$k$ objects defining them as the centroids of the clusters, and then, repeatedly does, for
each object, place the object inside the cluster to whose centroid it is closest,
calculating again the centroids for the cluster which has gain the object and also for
the cluster which has lost the object. After that, the algorithm repeats this last step
until there is no change in clusters’ composition between two consecutive iterations.
The most common and perhaps the best use of this algorithm requires the previous
knowledge of the number of classes to split the initial set of objects, which is the $k$
number.

In our experiments we intended to find the best $k$ number, in order to group the
dataset images into an optimal number of classes based on their features’ similarity.
Defining the best number of classes has been an open problem in recent times with a
considerable research activity and it can be validated using appropriate internal and/or
external criteria and techniques [12].

**Evaluation of clustering**

For the purpose of validating the clustering solutions and consequently assess the best
number of clusters we have ran the k-means algorithm varying $k$ from 3 to 20 and
then we used two different methods to evaluate the results. In one method
we computed some internal criteria for every $k$ tested while in the other method we
computed some external criteria. The internal criteria offer an idea of the solution
cohesion (how closely related are objects in a cluster) and also of the solution
separation (how well-separated a cluster is from other clusters) while the external
measures are related to how representative are the current clusters to the known
classes. They compare a clustering result with a known set of class labels to evaluate
the degree of consensus between the two. All the measures were calculated through the Cluster Validity Analysis Platform [13] in MATLAB.

It’s important to notice that all these results obtained by these methods provide some guidelines and do not indicate an exclusive “correct” number of clusters. They are at the disposal of the expert in order to evaluate the resulting clustering.

Thus, the first method that was used to validate the clustering solutions was based on the fact that it had no knowledge a priori, and therefore it could only be computed internal criteria. Among the criteria available, we chose to compute the Silhouette index [14] where the largest silhouette value indicates the optimal k, the Calinski-Harabasz index where also the maximum value indicates the optimal k, the C index [15] where the minimal C-index indicates the optimal k and finally the weighted inter-intra index that searches forward \((k=2,3,4,...)\) and stop at the first down-tick of the index, which indicates the optimal k.

On the second method, it was used the available image labeling information in order to evaluate the clustering solutions, becoming possible to compute external criteria. In this case there were computed the Jaccard index [16] where the maximum value indicates the optimal k, the Fowlkes-Mallows index [17] where also the maximum value indicates the optimal k and finally the Adjusted Rand index where also the maximum value indicates the optimal k.

The results of these methods are presented in the next section on this paper.

3 Results

In our experiments, we considered 13 classes of images from the MPEG-7 dataset and then we wanted to group them according to their seven previously extracted features, using the k-means clustering algorithm. So, the k-means algorithm was applied for k varying from 3 to 20, and then two methods were used to validate the clustering results. The first one computed some internal criteria, which are graphically represented in Figure 3 and the second one computed some external criteria that are in Figure 5.

As we can observe on Figure 3, only the C index suggests the number five as the optimal number of clusters, while the others indexes indicate the number six, which was actually the k number we considered. The following Table 1 expresses the images distribution among the six clusters based on the initial label information and Figure 4 illustrates three sample images that were grouped on the same cluster.
Figure 3. Graphical representation of the Silhouette index, the Calinski-Harabasz index, the $C$ index and the weighted inter-intra index.

Table 1. Distribution of the images through the six clusters versus the a priori partition (labels). Computed clusters are represented in columns and initial labels in rows.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>20</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bone</td>
<td>18</td>
<td>2</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cup</td>
<td>12</td>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>device3</td>
<td>15</td>
<td>5</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fork</td>
<td>15</td>
<td>5</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>guitar</td>
<td>1</td>
<td>19</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hammer</td>
<td>8</td>
<td>12</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>heart</td>
<td>1</td>
<td></td>
<td></td>
<td>19</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>horseshoe</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>key</td>
<td>19</td>
<td>1</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spoon</td>
<td>8</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>48</td>
<td>54</td>
<td>57</td>
<td>25</td>
<td>48</td>
</tr>
</tbody>
</table>
Figure 4. This figure shows three images that were grouped in the same cluster (cluster 4). Each image has on its top a radar plot graph of their features, represented with \( f_1, f_2, \ldots, f_7 \). With this type of representation it is noticeable how the image features shapes are quite similar and therefore grouped together.

The next Figure 5 illustrates the graphical representation of the external criteria computed within the second method. In this case the suggested value for the best \( k \) was the fourteen.

Figure 5. Graphical representation of the Jaccard index, the Fowlkes-Mallows index and the Adjusted Rand index.
To better understanding how images were clustered in respect to their initial labels, Table 2 illustrates the distribution among the fourteen clusters versus initial label information. As it can be observed, there are some clusters that perfectly match the initial label, as for instance the horseshoe and horse. Some other images are well concentrated in only one cluster, such as the key, bone, heart and apple. Some other images are grouped with images having different labels. An example is the spoon set of images that results distributed among four different clusters (C5, C9, C10 and C11). An observation of the images grouped within these clusters suggests that the spoon images are clustered with visually similar images (e.g. guitar and key).

Table 2. Distribution of the images through the fourteen clusters versus the a priori partition (labels). Computed clusters are represented in columns and initial labels in rows.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>bell</td>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>horse</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cup</td>
<td></td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>device</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td>15</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>shelf</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td>7</td>
<td>2</td>
<td>1</td>
<td></td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>hammer</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>heart</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>horse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>horseshoe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>spoon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>20</td>
<td>24</td>
<td>42</td>
<td>13</td>
<td>20</td>
<td>5</td>
<td>11</td>
<td>22</td>
<td>27</td>
<td>16</td>
<td>18</td>
<td>17</td>
<td>13</td>
</tr>
</tbody>
</table>

The used set of evaluation criteria, internal and external, does not suggest the same number of clusters. As the ultimate goal of this work is not to achieve a solution that recovers the truth (in form of the initial labeling), we consider from the conducted results analysis that the set of extracted features and selected algorithm is able to cluster images, in an unsupervised manner, in respect to their visual content.

We notice, that if some hint (a priori knowledge) about the initial number of classes would be known, as for instance in the form of a narrower range of expected clusters (e.g. from 10 to 16), the selected cluster would better match the initial partition, as the internal criteria would suggest a similar number of clusters. It is the case for the silhouette index, illustrated in Figure 3, which for that range suggests the 14 as the best number of clusters.

4 Conclusions and Future Work

In this paper, we presented a process to enable grouping of binary images by means of a learning technique over a set of extracted features. We assume that the process can be conducted without or with only few a priori information. The chosen
learning technique was the k-means clustering algorithm that needs the input of number of clusters (k). As, in our experience design, the number of clusters is not available or unrevealed on purpose, several clustering iterations were conducted for a wide range of the k value. Internal criteria were then used to choose the best number of clusters among all experiments. Results were then compared with the a priori partition in order to obtain an objective assessment of the ability of the unsupervised clustering match the represented object labeling.

Given the dataset, selected features and clustering algorithm, internal criteria suggests a best choice would occur for k = 6 clusters. The comparison with the (meanwhile revealed) initial partition, suggests a higher number of clusters, namely k = 14. This may indicate that the selected features are not enough or the more appropriate to obtain a clustering that closely matches the a priori partition. However, as illustrated, the images are closely grouped in respect to their visual silhouette. We notice that this is not an undesired result as the ultimate goal is not to match the given initial partition. The goal is to group images in a way that they can be indexed by their content (namely their visual properties). The fewer number of clusters suggest a reduction of the number of original concepts (initial labels) which is a feature/goal of this kind of clustering algorithm.

Achieved results are encouraging and suggest adequacy of the selected features and algorithm in order to group images by their visual content. Despite the former, we intend to explore new features and conduct an analysis to refine their selection. Instead of using the Euclidean distances to measure similarity we also consider the use of weighting attributes, according to their relevance.

We also intend to apply a supervised method (e.g.: k-nearest neighbor) to develop an information retrieval system. Using a query image, the system will extract the image features and classify it according to the sets previously defined with our clustering model. Then, it will return a set of images ranked according to similarity measures. Eventually this process could be evaluated by an expert and therefore it could provide more information/knowledge to the system. These are some of the future goals to improve our work presented.

References

7. MPEG-7 Core Experiment CE-Shape-1 Test Set, http://www.imageprocessingplace.com, 2009.12.01