Voting Clustering and Key Points Selection

Costas Panagiotakis[1] and Paraskevi Fragopoulou[2]

[1] Dept. of Commerce and Marketing, Technological Educational Institute (TEI) of Crete, 72200 Ierapetra, Greece,
cpanag@staff.teicrete.gr,
[2] Dept. of Applied Informatics and Multimedia, TEI of Crete, PO Box 140, Greece,
fragopou@ics.forth.gr**

Abstract. We propose a method for clustering and key points selection. We have shown that the proposed clustering based on the voting maximization scheme has advantages concerning the cluster's compactness, working well for clusters of different densities and/or sizes. Experimental results demonstrate the high performance of the proposed scheme and its application to video summarization problem.

Keywords: Clustering; Grouping; K-means; Video summarization

1 Introduction

Clustering is one of the most fundamental problems of pattern recognition with many applications in different fields like computer vision, signal-image-video analysis, multimedia, networks and biology. The clustering task involves grouping N given objects (points of $d-$dimensional space) into a set of K subgroups (clusters) in such a manner that the similarity measure between the objects within a subgroup is higher than the similarity measure between the objects from other subgroups [1]. Clustering algorithms can be divided into two main categories: hierarchical and partitional [2]. Hierarchical clustering algorithms recursively find nested clusters either in agglomerative (bottom-up) mode or in divisive (top-down) mode. According to partitional clustering algorithms, the clusters are simultaneously computed as a partition of the data. The resulting clusters can be disjoint and nonoverlapping (crisp clustering), where an object belongs to one and only one cluster, or overlapping (fuzzy clustering), where an object may belong to more than one cluster.

During the last decades, thousands of clustering algorithms [2] have been published, so hereafter we briefly present some popular and widely used clustering algorithms. An extensive survey of various clustering algorithms can be found in [2]. The K-means clustering algorithm, is one of the simplest partitional clustering algorithms that solves the clustering problem for a given number of clusters. The goal of K-means is to minimize the sum of squared error (SSE)

** Paraskevi Fragopoulou is also with the Foundation for Research and Technology-Hellas, Institute of Computer Science, 70013 Heraklion, Crete, Greece.

over all clusters. In [3], a variant method (K-means++ algorithm) for centroid initialization has been proposed that chooses centers at random from the data points, but weights the data points according to their squared distance from the closest center already chosen. K-means++ usually outperforms K-means in terms of both accuracy and speed. A deterministic initialization scheme for K-means is given by the KKZ algorithm [4]. According to KKZ method, the first centroid is given as the data point with maximum norm, and the second centroid is the point farthest from the first centroid, the third centroid is the point farthest from its closest existing centroid and so on. An extension/variation of K-means is the K-medoid or Partitioning Around Medoids (PAM) [5], where the clusters are represented using the medoid of the data instead of the mean. Medoid is the object of the cluster with minimum distance to all others objects in the cluster. Most of the approaches from literature are heuristic or they try to optimize a criterion that may not be appropriate for clustering or they require a training set. On the contrary, in this paper, we have solved the crisp clustering problem via a voting maximization scheme that ensures high similarity between the points of the same cluster without any user defined parameter. In addition, the proposed method has been applied to video summarization problem [6].

2 The Clustering Problem

In this section the clustering problem is analyzed. Let us assume a dataset of N points, $x_i, i \in \{1, ..., N\}$, in the d dimensional space ($x_i \in \Re^d$) that are clustered into K non empty clusters, $p_k, k \in \{1, ..., K\}$, where p_k denotes the k-cluster indexes and $|p_k|$ denotes the number of points of cluster p_k. According to crisp clustering it holds that each point belongs to exactly one cluster.

One of the most widely used criteria for clustering and for other similar problems (e.g. see Microaggregation problem [7]) is the within-group squared error (SSE) minimization,for cases of almost equal sized clusters and almost the same variation, the minimization of SSE yields what the humans mean "optimal clustering". However, the clustering that corresponds to the minimization of SSE is not always appropriate even for the simple case of two clusters. According to the minimization of SSE, it is difficult to keep connected large clusters with high variation, that means that if there exists a large physical cluster with high variation it is possible to be divided into two or more clusters.

In this research, we introduce a new validity measure, the Voting Measure (VM) that can also work well for clusters with different densities and/or sizes. VM is invariant on scaling and number of data points and is bounded $VM \in [0, 1]$. In order to define VM, first we introduce the voting point problem. According to this problem, we have to define the function $V(i, j) \in [0, 1]$ that corresponds to the votes of point $x_i,\ i \in \{1, ..., N\}$ to point $x_j, j \in \{1, ..., N\}$. However, if we use a metric for points' density like the Gaussian similarity function in spectral clustering, then high density clusters will be favored. In order to overcome this problem, the voting function is defined so that it should satisfy the following conditions:

(a) $\sum_{j=1}^{N} V(i,j) = 1$, **(b)** $V(i,i) = 0$, **(c)** $V(i,j) \sim \frac{1}{d(x_i,x_j)}$, **(d)** $V(i,j) \leq \frac{1}{2}$ where $d(x_i, x_j)$ denotes the Euclidean distance between the points x_i, x_j. The first two conditions ensure the point "equality" (each point have the same voting "power"). The third condition ensures the scale/density invariant property. According to the first three conditions it holds that $V_3(i,j) = \frac{\frac{1}{d(x_i,x_j)}}{\sum_{k\in\{1,\ldots,N\}-\{i\}} \frac{1}{d(x_i,x_k)}}$, where $V_3(i,j)$ denotes the voting matrix that satisfy the first three conditions (the sub-index show the number of satisfied conditions). The last condition is added in order to ensure that each point is will vote the rest points, avoiding the special case of pairs of identical points that only vote each other resulting wrong voting descriptors (see at the end of the Section). When all the conditions are satisfied then $V_4(i,j)$ is given by:

$$V_4(i,j) = \begin{cases} V_3(i,j) & , \delta(i) \leq 0 \\ \min(\frac{V_3(i,j)}{1-\delta(i)}, \frac{1}{2}) & , \delta(i) > 0 \end{cases} \quad (1)$$

where $\delta(i) = \max_{j\in\{1,\ldots,N\}} V_3(i,j) - \frac{1}{2}$. In our experimental results, the voting matrix is computed based on the four prementioned conditions. The voting descriptor $VD(j) = \sum_{i=1}^{N} V(i,j)$ of point x_j, $j \in \{1,...,N\}$ measures the votes that point x_j receives. Under any dataset, it holds that the mean value of VD is one ($E(VD) = 1$). VM is defined by the average value of voting descriptors per cluster taking into account only the intrinsic voting, dividing by the number of clusters K:

$$VM = \frac{1}{K} \cdot \sum_{k=1}^{K} \frac{\sum_{i\in p_k} \sum_{j\in p_k} V(j,i)}{|p_k|} \quad (2)$$

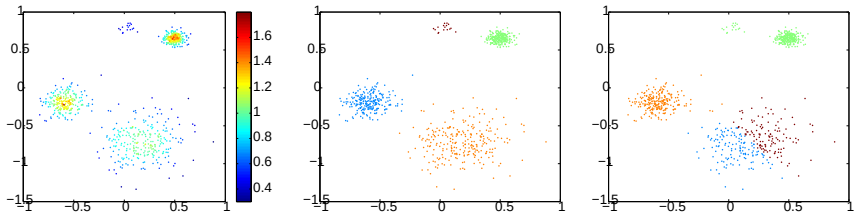
Fig. 1(a) depicts a dataset using a colormap according to voting descriptor (red for high values and blue for low values). It holds that the voting descriptor generally receives higher values on points that are closer to a cluster centroid, while it receives lower values on boundary points. Lower values (e.g. close to zero) are observed for outliers, since these points are quite far from clusters, thus it is difficult to receive votes.

3 The Proposed Algorithm

3.1 Voting-based Clustering Algorithm

In this section, the proposed Clustering based on Voting Representativeness algorithm (CVR) is presented. This method requires as input the voting array V, the voting descriptor VD, and the K. The output of the method is the cluster indexes. The proposed Clustering based on Voting Representativeness algorithm (CVR) method consists of two phases:

- In the first phase, K iterations are performed selecting the K key points. In the k^{th}-iteration of the method, we select a key point of the dataset to be the representative of the k-cluster and we discard it from the dataset.



(a) (b) (c)

Fig. 1: **(a)** The dataset using a colormap according to voting descriptor. Results of clustering **(b)** $K = 4$, $SSE = 27.68$ and **(c)** $K = 4$, $SSE = 23.89$.

Therefore, at the end of the first phase, the each cluster has been initialized with one record.

– Finally, in the second trivial phase the $N - K$ remaining unlabeled points are assigned to the cluster that corresponds to their "closest" representative point according to the voting formulation.

Hereafter, we analyze the first phase of the proposed method with more details, that has been used in video summarization problem (see Section 4). In the first iteration of the first phase, we detect the most representative (key) point of the dataset (p_1), where VD is maximized. This key point will be assigned to the first cluster and it will be discarded from the dataset (S).

For the selection of the next key points $p_k, k > 1$, we have to taken into account the already selected representative points. Therefore, the second key point ($p_k,\ k = 2$) is selected taking into account the first one. This point should belong to a different cluster meaning that it should have low similarity with p_1 and vice versa. In order to satisfy this condition, we select the point with index i that minimizes the formula $v = \frac{V(i,p_{k-1})}{VD(p_{k-1})} + \frac{V(p_{k-1},i)}{VD(k)}$. This is the sum of percentages of votes that the point with index p_1 receives from the point with index i and vice versa.

We initialize a function $F(i) = 0$, $i \in S$. The next key points are selected by repeating the same procedure using the function F. When a point (x_{p_k}) is selected as a key point, we add it to the appropriate cluster and we discard it from the set S, where S denotes the domain of F. Finally, F is updated in order to ensure that the next key points will have low similarity with the already computed key points $F(i) = \max(F(i), v)$ as well as with x_{p_k}. The global minima of F will give the next key points. The total computational cost of the proposed CVR algorithm can be reduced from $O(N^2)$ to $O(N \cdot \log N + K \cdot N)$ when a sparse matrix and R-tree-like data structure are used.

3.2 Local Maximization of VM

This section presents an optional algorithm, inspired by the GSMS-T2 [7], that possibly improves a given initial clustering based on the local maximization of Voting Measure (VM). When we use as input the clustering of CVR method, the resulting algorithm is called CVR-LMV. Let's assume that two nearby located points $x_i, x_j,\ i, j \in \{1, ..., N\}$ that are misclassified by CVR in the same cluster, so that $V(i,j) \geq V(i,k), \forall k \in \{1, ..., N\}$ and $V(j,i) \geq V(j,k), \forall k \in \{1, ..., N\}$. Under this assumption, it is possible that if we separately check to reassign the point i (or the point j) to the true cluster, VM will be reduced, since the point x_j (or the point x_i) belongs to a different cluster.

In order to solve this problem without increasing the computation cost of the algorithm, we have introduced the median based VM $\widehat{VM}$, that estimates VM based on the median value of votes of points without affected by nearby points.

$$\widehat{VM} = \frac{1}{K} \cdot \sum_{k=1}^{K} \sum_{i \in p_k} median_{j \in p_k}(V(j,i)) \tag{3}$$

Let VM and VM' denote the validity measure before and after the possible reassignment. Let $\widehat{VM}$ and $\widehat{VM}'$ denote the median based VM before and after the possible reassignment. According to the proposed algorithm, we reassign the point with index i, if $VM' > VM$ or $\widehat{VM}' > \widehat{VM} \wedge \widehat{VM}' - \widehat{VM} + VM' - VM > 0$ is satisfied. The first condition ensures that VM increases. If only the second condition is true, this will cause an impermanent decrease of VM. Since the increase of $\widehat{VM}$ is higher than the decrease of VM means that the point with index i is closer to the examined cluster and we have to perform reassignment. In the next steps, we will also reassign the neighbors of point with index i and VM will increase.

Figs. 1(b) and 1(c) illustrate two different clustering results of the same dataset using the CVR-LMV and K-means clustering, respectively. The SSE of clustering depicted in Fig. 1(c) is 13.69% lower than the SSE of clustering depicted in Fig. 1(b). However the optimal solution of clustering is clearly depicted in Fig. 1(b).

4 Experimental Results

In this section, the experimental results of our performance study are presented. We have tested our methods (CVR and CVR-LMV) using the following six real datasets [8], where the number of records, the number of clusters, the data dimension, the cluster sizes and cluster densities are varied:

- the Iris (150 records in 4-dimensional space, $K = 3$).
- the Yeast (1484 records in 8-dimensional space, $K = 10$).
- the Segmentation (2100 records in 19-dimensional space, $K = 7$),
- the Wisconsin breast cancer (683 records in 30-dimensional space, $K = 2$).

– the Wine (178 records in 13-dimensional space, $K = 3$).
– the first 10^4 records of covtype (covtype10k) in 54-dimensional space, $K = 7$.

We have tested the proposed methods with 144 synthetic datasets generated by c random cluster centroids that are uniformly distributed over the d-dimensional hypercube ($c \in \{4, 8, 16\}$, $d \in \{4, 8\}$). The number of points n_i in cluster i is randomly selected from a uniform distribution between $\min n$ and $\max n$ ($\min n \in \{16, 128\}$, $\max n - \min n \in \{0, 128\}$). The n_i points in cluster i are randomly selected around the cluster centroid from a d-dimensional multivariate Gaussian distribution with covariance matrix $\Sigma_i = \sigma_i^2 I_d$ and mean value equal to the cluster centroid, where σ_i is randomly selected from a uniform distribution between $\min \sigma$ and $\max \sigma$, ($\min \sigma \in \{0.04, 0.08, 0.16\}$), ($\max \sigma - \min \sigma \in \{0, 0.08\}$). The parameters c and $\min \sigma$ receive three different values and the rest of the parameters receive two different values yielding $3^2 \cdot 2^4 = 144$ datasets.

In order to evaluate the accuracy of the proposed scheme, we have compared the proposed methods with seven other clustering methods: the K-means, the K-means KKZ algorithms [4], the hierarchical agglomerative algorithm based on the linkage metric of average link (HAC-AV) [9], spectral clustering using Nystrom method without orthogonalization (SCN) and with orthogonalization (SCN-O) [10], the K-means++ method [3] and the PAM algotithm [5]. For the non deterministic algorithms, 20 trials have been performed under any given dataset, getting the average value of the used performance metrics. We evaluate the performance using the clustering accuracy (Acc) [10]. $Acc \in [0, 1]$ is defined as the percentage of the correctly classified points.

Dataset	**CVR-LMV**	**CVR**	**K-means**	**K-means KKZ**	**HAC-AV**	**SCN**	**SCN-O**	**PAM**	**K-means++**
Iris	**93.33%**	81.33%	84.20%	89.33%	90.67%	89.10%	88.87%	77.43%	85.77%
Yeast	39.22%	**42.39%**	36.04%	37.80%	32.35%	37.54%	37.10%	32.37%	35.15%
Segmentation	52.14%	37.43%	51.87%	35.62%	14.62%	47.35%	46.55%	**52.45%**	50.86%
Wisconsin	**91.04%**	90.51%	85.41%	85.41%	66.26%	73.15%	85.14%	84.97%	85.41%
Wine	**71.35%**	**71.35%**	68.20%	56.74%	61.24%	66.04%	60.17%	67.44%	65.65%
covtype10k	37.10%	**38.18%**	36.41%	35.95%	35.63%	36.20%	36.49%	35.90%	36.97%
144 S.D.	**98.71%**	97.85%	79.51%	97.51%	97.01%	94.04%	97.08%	78.61%	86.21%

Table 1: The accuracy (first 6 lines) and the average Acc (last line) of several clustering algorithms in 6 real and 144 synthetic datasets (144 S.D.), respectively.

Table 1 depicts the clustering accuracy measure of CVR-LMV, CVR, K-means, K-means KKZ, HAC-AV, SCN, SCN-O, PAM and K-means++ algorithms in real datasets (first six lines of the table) and the average clustering accuracy measure over the 144 synthetic datasets (144 S.D.) (last line of table). According to these results, the proposed methods CVR-LMV and CVR yield the highest performance results, outperforming the other methods from literature in five out of six real datasets. The highest performance results are achieved by CVR-LMV, since it holds that almost always, it gives the highest or the second highest performance results. According to the experiments on synthetic datasets, CVR-LMV yields the highest performance results, outperforming the other algorithms. CVR is the second highest performance method. High performance results are also obtained by K-means KKZ, HAC-AV and SCN-O methods.

Concerning the probability that CVR-LMV reduces the clustering performance, this probability increases when CVR fails to find the true classes. In this case, CVR-LMV is possible to reduce or increase the clustering performance.

Fig. 2: Selected key frames of tennis (**(a),(b),(c)**) and foreman (**(e),(f)**) videos.

Fig. 3: Selected key frames of hall monitor video.

The proposed method can be used on several clustering based applications like the video summarization using key frames [6], where the goal is to select a subset of a video sequence (key frames) that can represent the video visual content. Similarly to [6], we have used the Color Layout Descriptor (CLD) which suffices to describe smoothly the changes in visual content. Then we apply the CVR algorithm using as input the CLD vectors and the desired number of key points K. The key points of the first phase of the CVR algorithm can be considered as the selected key frames, since they have the property to cover the video content space belonging to different clusters according to the CVR algorithm. An advantage of the proposed method is that ordering of the resulting key frames corresponds to their significance. Moreover, the proposed method does not assume that the video file has been segmented into shots as most of the key frame extraction algorithms done. The proposed method has been tested in several indoor and outdoor real life video sequences that have been used in [6] describing well the video content. Hereafter, we present the results of the proposed method on tennis, foreman and hall monitor videos[3] (see Figs. 2, 3) using three, two and five key frames, respectively. Under any case, it holds that the selected key frames are close to the humans' perception: In the tennis video, the first two selected key frames (#274, #120) belong to the two different shots of the video and the third one (#20) belongs on the first shot that has substantial visual content changes. In the foreman video, the selected key frames belong on the start and end of the sequence, describing well the two characteristics phases of the sequence (the interview and the buildings). In the hall monitor video the

[3] http://media.xiph.org/video/derf/

five selected key frames correspond to the five different "scenes" of the video (empty hall, a human with a bag in hall and so on).

5 Conclusions

In this paper, we propose a deterministic point clustering method that can be also used in video summarization problem. According to the proposed framework, the problem of clustering is reduced to the maximization of the sum of votes between the points of the same cluster. In addition, we have proposed the LMV algorithm that possibly improves a given initial clustering based on the local maximization of the proposed robust voting measure (VM). The proposed method can yield high performance results on clusters of different densities and/or sizes outperforming other methods from literature. In addition, the selected key frames describes well the visual content of the videos.

Acknowledgments

This research has been partially co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program ARCHIMEDE III-TEI-Crete-P2PCOORD.

References

1. Gupta, U., Ranganathan, N.: A game theoretic approach for simultaneous compaction and equipartitioning of spatial data sets. IEEE Transactions on Knowledge and Data Engineering **22** (2010) 465–478
2. Jain, A.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters **31** (2010) 651–666
3. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. (2007) 1027–1035
4. Katsavounidis, I., Kuo, C.C.J., Zhang, Z.: A new initialization technique for generalized lloyd iteration. IEEE Signal Processing Letters **1** (1994) 144–146
5. Theodoridis, S., Koutroumbas, K.: Pattern Recognition 3rd edition. Elsevier (2006)
6. Panagiotakis, C., Doulamis, A., Tziritas, G.: Equivalent key frames selection based on iso-content principles. IEEE Transactions on Circuits and Systems for Video Technology **19** (2009) 447–451
7. Panagiotakis, C., Tziritas, G.: Successive group selection for microaggregation. IEEE Trans. on Knowledge and Data Engineering **99** (2011 (accepted))
8. Blake, C., Keough, E., Merz, C.J.: UCI Repository of Machine Learning Database (1998) URL: `http://www.ics.uci.edu/~mlearn/MLrepository.html`.
9. Day, W., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of classification **1** (1984) 7–24
10. Chen, W.Y., Song, Y., Bai, H., Lin, C.J., Chang, E.Y.: Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 568–586