Ssian Journal of Information Technology 61-67, 2004
Special Issue on Machine Learning in China - Selected Papers of CCML'04

A Possibilistic C-Patterns Algorithm and its Application in Video Abstraction

[1]Xinbo Gao, [1]Jie Li and [2]Halidan, A.
[1] School of Electronic Engineering, Xidian University, Xi'an, China
[2] School of Electrical Engineering, Xinjiang University, Urumuqi, China

Abstract: Cluster analysis is an important unsupervised machine learning technique and has been widely applied in the field of pattern recognition, data mining and computer vision. As a popular clustering method, fuzzy *c*-means algorithm partitions any a specified pattern sample set into clusters and reveals the degrees of sharing for each pattern sample in all the clusters. To find the meaningful clusters as defined by dense regions rather than to generate partition, the possibilistic *c*-means (PCM) was proposed as a mode- seeking algorithm to cast the clustering problem into the framework of possibility theory. The PCM algorithm assigns and interprets the membership degree as typicality or compatibility. However, each clustering prototype obtained by the PCM algorithm is the possibilistic mean of all the patterns in the corresponding cluster, which sometimes does not correspond to a real existing pattern in the data set. To extract the real typical pattern as clustering prototype for each cluster, a novel possibilistic *c*-patterns (PCP) algorithm is presented in this paper. Like the PCM algorithm, the PCP algorithm is also a mode-seeking clustering algorithm. However, it can be used to search the representative samples as prototypes for all the obtained clusters. Based on the favorable characteristics of the PCP algorithm, it is applied to video abstraction for content-based video retrieval. The experimental results with synthetic data and real video data illustrate the effectiveness of the typical-pattern seeking from noise contaminated data set as well as the key frame extraction of video sequence.

Key words: Cluster analysis, possibilistic *c*-patterns algorithm, video abstraction

Introduction

As well known, cluster analysis is an effective approach to multi-variant statistical analysis, and one of important branches of unsupervised machine learning. So far, it has been widely used in the fields of data mining, knowledge database discovery, computer vision, and content-based information retrieval. Most traditional clustering methods are based on crisp partition, in which each pattern (sample) is strictly classified into one and only one of the groups. Hereby, the boundaries among groups are sharply and deterministic. Since the practical memberships of most patterns are not strict but ambiguous, hence it is suitable for soft classification (Xinbo Gao, Weixin Xie, 2000). With the advent of fuzzy sets theory (Zadeh, 1965), possibility theory (Dubois and Prade, 1988), and rough sets theory (Pawlak, 1982), they were introduced into the traditional cluster analysis to form a family of soft clustering algorithms. Due to the soft clustering can reveal the degree of uncertainty for each pattern in any group and describe the medium property of patterns' membership, it reflects the real world more objectively and exactly. Thereby, the soft cluster analysis has become the main research topic of machine learning.

It was Ruspini who first introduced the concept of fuzzy partition into cluster analysis and extended the objective function J_1 of the hard *c*-means algorithm to J_2 for the fuzzy clustering case (Ruspini, 1969). Then, Bezdek generalized the fuzzy objective function into an infinite family $J_m(1 \le m < \infty)$ by introducing a fuzzy weighting exponent *m* (Bezdek, 1981). By an alternate optimization (AO) strategy, the objective function J_m is minimized, in which the AO strategy is the famous fuzzy *c*-means (FCM) clustering algorithm. Finally, the FCM algorithm outputs the fuzzy partitioning result for the given pattern sample set by assigning each pattern a membership degree in any cluster. In this way, the FCM algorithm is often called a partitioning algorithm. However, for the probabilistic constraint of fuzzy membership degree, the membership degree generated by the FCM algorithm is interpreted as degree of sharing of each pattern among all the clusters but not as degree of typicality. This gives rise to poor performance in the presence of noise and outliers. Moreover, given any a cluster number *c* the FCM algorithm always partitions the specified pattern set into *c* fuzzy subsets, regardless of how many real "clusters" are actually contained in the pattern set. In other words, each subset may or may not correspond to a real "clusters".

To this end, a modification to the FCM objective function was made by relaxing the probabilistic constraint of membership degree, and a new clustering algorithm, namely the possibilistic *c*-means (PCM) algorithm, was derived (Krishnapuram and Killer, 1996). In the PCM algorithm, the membership value of a pattern in a cluster (or class) represents the typicality of the pattern in the class, or the possibility of the sample belonging to the class. Since noise patterns or outliers are less typical, typicality-based memberships automatically reduce the effect of noise patterns and outliers, and improve the classification results considerably. The power of the PCM algorithm does not lie in creating partition, but rather in finding meaningful clusters as defined by dense regions.

As a by-production, the FCM and PCM algorithms also generate a cluster prototype for each cluster as a representative pattern of each class. Actually, the cluster prototype of a class is the fuzzy or possibilistic mean of all the patterns in each cluster. Since the membership value of the FCM represents the degree of sharing rather than the degree of typicality. It is obvious that the computed cluster centroid cannot be looked as typical pattern of this cluster. In the other hand, although the PCM is primarily a mode-seeking algorithm and the prototypes are automatically attracted to dense regions in feature space with the iterative process, in some cases, the possibilistic centroid of each cluster also cannot be regarded as representative pattern of the cluster. For instance, the pattern space is discontinuous with finite states like categorical (Boolean, or words) attributes data, the probabilistic (or possibilistic) mean of all the patterns in a cluster does not correspond to a real existing sample (or pattern), and is a meaningless prototype. For the application of both classification and the typical patterns selection for a give pattern set, the PCM does not work as well as FCM. For this purpose, this paper presents a novel typical pattern extraction algorithm, namely the possibilistic *c*-patterns (PCP), by a minor modification for the PCM algorithm. Besides inheriting the advantages of the PCM algorithm, the PCP algorithm can also extract the meaningful typical pattern for each cluster. Based on the favorable characteristics of the PCP algorithm, it is applied to video abstraction for content-based video retrieval with good performance.

Soft *C*-Partition and Soft Clustering Algorithms

Soft C-Partition of Pattern Set: Let $X = \{x_1, x_2, ..., x_n\}$ be a finite pattern sample set. For each pattern, $xk = (x_{kl}, x_{k2}, ..., x_{kp})^T \in R^P$ is a feature vector, and x_{kj} is the *j*-th characteristic of pattern x_k. Cluster analysis refers to partition the given pattern set *X* into *c* non-overlapped subsets $X_1, X_2, ..., X_c$ according to the similarities among the given patterns, which meets the following conditions.

$$X_1 \cup X_2 \cup ... \cup X_c = X;\ X_i \cap X_j = \varnothing,\ 1 \le I \ne j \le c;\ X_i \ne \varnothing,\ X_i \ne X \tag{1}$$

The membership of each pattern x_k $(1 \le k \le n)$ to subset X_i $(1 \le i \le c)$can be denoted by the following membership function.

$$\mu_{X_i}(x_k) = \mu_{ik} = \begin{cases} 1, & x_k \in X_i \\ 0, & x_k \notin X_i \end{cases} \tag{2}$$

Since $\mu_{ik} \in \{0, 1\}$, that is to say, each pattern can belong to and only belong to a certain subset, and the obtained partition is called a crisp *c*-partition. The obtained partition space in this way is defined as crisp *c*-partition space E_h.

$$E_h = \left\{ U \in R^{c \times n} \middle| \mu_{ik} \in \{0,1\}; \sum_{i=1}^{c} \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^{n} \mu_{ik} < n, \forall i \right\} \tag{3}$$

Ruspini extended the membership function from {0,1} to [0,1] with the fuzzy sets theory, so that the crisp *c*-partition was generalized into a fuzzy *c*-partition E_f.

$$E_f = \left\{ U \in R^{c \times n} \middle| \mu_{ik} \in [0,1]; \sum_{i=1}^{c} \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^{n} \mu_{ik} < n, \forall i \right\} \tag{4}$$

For the fuzzy partition, if the probabilistic constraint $\sum_{i=1}^{c} \mu_{ik} = 1, \forall k$ is relaxed, the fuzzy partition will be evolved to the possibilistic partition, which is defined as E_p.

$$E_p = \left\{ U \in R^{c \times n} \middle| \mu_{ik} \in [0,1]; \max_{i=1}^{c} \mu_{ik} > 0, \forall k;\ 0 < \sum_{k=1}^{n} \mu_{ik} < n, \forall i \right\} \tag{5}$$

Obviously, the *c*-dimensional possibilistic labeling vector $\mu_k = [\mu_{1k}, \mu_{2k}, ..., \mu_{ck}]^T$ takes value in the *c*- dimensional unit hypercube. The fuzzy labeling vector μ_k takes value on the hyper-plane cross the *c* unit base vectors, and crisp labeling vector μ_k takes value on the *c* unit base vectors in the real *c*-hypercube.

FCM Algorithm: To realize the crisp *c*-partition of patterns set, the cluster analysis is converted to a nonlinear programming problem with constraint as follows.

$$\min J_1(U,P)=\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}\cdot\|x_k-p_i\|_A^2,\quad s.t.\ U\in E_h \tag{6}$$

Where $U=[\mu_{ik}]_{c\times n}$ is the partition matrix, and denotes the cluster prototype matrix, in which $P_i\in R^p$ indicates the prototype vector of the *i*-th cluster. In addition, $\|.\|^2_A$ is a kind of matrix weighting norm to evaluate the dissimilarity (distance) between the pattern x_k and the prototype p_i. Here, *A* is a symmetrical positive definite matrix. In case of A = I (I is unit matrix), the dissimilarity becomes the Euclidean distance in pattern space.
In 1973, Dunn extended the within group sum of squared error (WGSSE) function $J_1(U, P)$ to the weighted WGSSE function $J_2(U, P)$ to solve the fuzzy clustering problem. Then, Bezdek further generalized $J_2(U, P)$ to a family of objective functions, $J_m(U, P)$.

$$\min J_m(U,P)=\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^m\cdot\|x_k-p_i\|_A^2,\quad m\in[1,+\infty)\quad s.t.\ U\in E_f \tag{7}$$

Where *m* is called weighting exponent, or smooth parameter, which controls the fuzziness of cluster analysis result. Like the hard *c*-means (HCM) algorithm, the fuzzy *c*-means (FCM) algorithm is also an alternate optimization strategy. By the iterative update between the membership function μ_{ik} and the cluster prototype p_i as shown in Eq.(8) and Eq.(9), the objective function Eq.(7) converges into a local minimum. Meanwhile, the FCM algorithm obtains the optimal *c*-partition of the given pattern set.

$$\mu_{ik}^{(t)}=\left\{\sum_{j=1}^{c}\left[\left(\frac{\|x_k-p_i^{(t)}\|_A}{\|x_k-p_j^{(t)}\|_A}\right)^{\frac{2}{m-1}}\right]\right\}^{-1},\quad i=1,2,\cdots,c,\ k=1,2,\cdots,n \tag{8}$$

$$p_i^{(t+1)}=\frac{1}{\sum_{k=1}^{n}\left(\mu_{ik}^{(t)}\right)^m}\sum_{k=1}^{n}\left(\mu_{ik}^{(t)}\right)^m x_k,\quad i=1,2,\cdots,c,\ k=1,2,\cdots,n \tag{9}$$

PCM Algorithm: Fuzzy *c*-means clustering is one of partitioning algorithms, which can realize the fuzzy *c*-partition for the given pattern set. As a by-production, the FCM algorithm obtains a cluster prototype for each group, i.e., the fuzzy centroid (mean) of each cluster. However, due to the existence of probabilistic constraint $\sum_{i=1}^{c}\mu_{ik}=1$, $\forall k$, the obtained membership degree of each pattern to any group cannot represent the typicality or compatibility. In the case of contaminated pattern set with noise, the obtained prototype may be disturbed and cannot serve as the representative pattern of the corresponding cluster. To suppress the noise interference and extract typical prototypes, by relaxing the probabilistic constraint of the FCM membership degrees, Krishnapuram and Killer (1996) proposed the possibilistic *c*-partition concept and designed the possibilistic *c*-mean (PCM) clustering algorithm.
In the PCM algorithm, the objective function of clustering is redefined as follows.

$$\min J_m(U,P)=\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^m\cdot\|x_k-p_i\|_A^2+\sum_{i=1}^{c}\eta_i\sum_{k=1}^{n}(1-\mu_{ik})^m,\quad m\in[1,+\infty)\quad s.t.\ U\in E_p \tag{10}$$

In Eq.(10), the η_i defines a valid region of the *i*-th group, which can be updated with Eq.(11).

$$\eta_i=\beta\cdot\frac{\sum_{k=1}^{n}\mu_{ik}^m\cdot\|x_k-p_i\|_A^2}{\sum_{k=1}^{n}\mu_{ik}^m},\quad i=1,2,\cdots,c \tag{11}$$

That is to say, the η_i is proportional to the average distance within a fuzzy group. In general, we take $\beta=1$. The update formula of μ_{ik} is given as

$$\mu_{ik}=\frac{1}{1+\left(\frac{\|x_k-p_i\|_A^2}{\eta_i}\right)^{\frac{1}{m-1}}},\quad i=1,2,\cdots,c,\ k=1,2,\cdots,n \tag{12}$$

The update formula of cluster prototype of PCM algorithm is given in Eq.(9), the same as that of FCM algorithm, i.e., the fuzzy mean of patterns for each group. Since the membership function generated by the PCM interprets the typicality, the resulted prototype will be of typicality. Of course, the computational complexity of the PCM algorithm is greater than that of the FCM. In the one hand, the parameter η_i should be recalculated in each iteration step. In the other hand, the PCM algorithm is sensitive to the initialization, so it usually uses the FCM result as initialization.

PCP Algorithm and Its Application in Video Abstraction: PCP Algorithm: As mentioned above, the probabilistic constraint causes the FCM algorithm to generate memberships that can be interpreted as degrees of sharing. By relaxing the constraint on memberships, one can generate memberships that represent typicality by the PCM algorithm. It was also shown that since noise patterns or outliers are less typical, typicality-based memberships automatically reduce the effect of noise patterns and outliers, and improve the clustering results considerably. In addition, the PCM can obtain more typical prototypes than the FCM algorithm. So, the PCM algorithm can also be employed to extract the representative key patterns for a given pattern set.
From the Eq.(9), it is observed that the prototype generated in PCM is the possibilistic mean of patterns in a cluster. In some applications, the possibilistic mean does not correspond to a real existing sample (or pattern) in pattern set, and even is a meaningless prototype. For instance, in the discontinuous (or discrete) pattern feature space like nonnumeric, Boolean or categorical attributes feature space, the possibilistic or fuzzy mean of patterns lacks of real meaning.
For this purpose, we propose the possibilistic *c*-patterns (PCP) algorithm by modifying the prototype update scheme of the PCM algorithm. In the PCP algorithm, the prototype for each group is selected as a pattern meeting the following condition in each iteration step.

$$p_i = x_r,\ \text{such that} \left| \left(\sum_{k=1}^{n} \mu_{ik}\right) x_r - \sum_{k=1}^{n} \mu_{ik} \cdot x_k \right| = \min_{l=1}^{n} \left| \left(\sum_{k=1}^{n} \mu_{ik}\right) x_l - \sum_{k=1}^{n} \mu_{ik} \cdot x_k \right| \qquad (13)$$

Since the representative pattern is taken as the cluster prototype, the PCP algorithm is very suitable for the applications of typical patterns extraction.
Compared to the FCM algorithm, the PCP algorithm has some advantages. First, the PCP is a mode- seeking algorithm rather than a partitioning algorithm, and it is good at finding meaningful clusters as defined by dense regions. Secondly, like the PCM algorithm, the PCP has advantage of noise-suppression over the FCM algorithm. Thirdly, as well known, in the FCM, the cluster number *c* should be specified in advance as an integer greater than 1. While, the PCP algorithm can sequentially remove clusters after repeatedly applying the PCP with $c = 1$. One can in fact determine the appropriate number of clusters, thus addressing the problem of cluster validity.
Compared to the PCM algorithm, the PCP algorithm has also some advantages. First, the PCP can generate not only typical memberships but also typical patterns. Secondly, the PCP can be applied in both continuous and discrete pattern space, especially, the data set with nonnumeric, Boolean or categorical attributes. Thirdly, since the search space is finite pattern states, the PCP algorithm has less computational complexity and better convergence than the PCM algorithm.

PCP-Based Video Abstraction: As digital video libraries and archives of immense size are becoming accessible over data networks, efficient video retrieval and browsing have become crucially important (Aigrain, Zhang, and Petkovie, 1996). Automatic video abstraction refers to processing of video source to extract description schemes and visual summaries to enable rapidly comprehending the content or coverage of an archived video collection, discerning the activities depicted by the collection, identifying clips of interesting activities, and previewing those clips to locate particular events. The visual summary form, called a video abstract, consists of a sequence of key frames (Gunsel and Teklap, 1998) or a mosaics (static content) and moving object trajectories (dynamic content) (Arthur Pope, Rakesh Kumar, Harpreet Sawhney, and Charles Wan, 1998.) that can represent the salient content of a video clips. The use of key frames greatly reduces the amount of data required in video indexing and browsing and provides an organizational framework for dealing with video content. Users can quickly browse over the video and search the interest video by viewing only a few highlighted key frames.
Current key-frame-extraction methods can be classified according to their various measurements of visual content complexity of a video shot or sequence and the criteria for determining the key frame. For the former, most measure metrics are based on the color histogram comparison (Gunsel and Teklap, 1998; Hanjalic and Zhang, 1999.), and others are based on the motion vectors, for example, Liu et al. (2003) uses the perceive motion energy model to measure the visual content complexity. With the measurement of the visual content difference, the approach to partitioning the video sequence into homogenous clusters can be divided into three classes, threshold-based method, floating-threshold method and threshold-free method. Obviously, a fixed threshold cannot fit for all the possible video sequence. Gunsel et al. (1998) apply the OTSU method to automatically determine the adaptive threshold. While, Hanjalic et al. (1999) use the unsupervised clustering method to complete the classification task without threshold. How to extract the key frames has corresponding three categories. The first method directly extracts the first and/or the middle and/or the last frame from each homogenous cluster as key frame(s) (Aigrain, Zhang, and Petkovie, 1996). The second method takes the clustering prototypes as key frames, and the number of key frames is automatically by cluster validity function (Hanjalic and Zhang, 1999). The last method extracts the frames at the turning point of the motion acceleration and motion deceleration as key frames (Tianming Liu, Hongjiang Zhang and Feihu Qi, 2003).

Obviously, a threshold-free key frames extraction method is preferred with the size of key frame set automatic determination. As mentioned above, the clustering prototypes sometimes cannot represent the typical pattern. Hereby, the FCM and PCM algorithms are not suitable for extracting the key frames from the video sequence. Here, the proposed PCP algorithm is employed to fulfill such a goal. The PCP algorithm can not only extract the typical key frames from the video sequence, but also can automatically determine the number of key frames. In addition, it can extract any size of key frame set as you wish according to the order of typicality degree.

Unlike most of the reported work in the literature, we propose an algorithm that performs scene change detection and key frame selection in one step by using a high dimensional feature vector. The scheme is based on color similarity analysis in the HSV space and the temporal continuous analysis, because color is a consistent characteristic of each shot.

In the HSV space, the value of hue, saturation and intensity are quantized with Q_h: Q_s: Q_v, say, 8 : 3 : 3. We extract the color histogram of HSV to synthesize a $Q_h + Q_s + Q_v$ dimensional feature vector for any frame, $H = (H_h(0),\ldots, H_h(Q_h-1), H_s(0),\ldots, H_s(Q_s-1), H_v(0),\ldots, H_v(Q_v-1))$. By adding the temporal axis, we perform the key frame extraction in the $Q_h + Q_s + Q_v + 1$ dimensional feature space with the proposed PCP clustering algorithm. For any a frame-pair (f_i, f_j), $i, j = 0, 1, \ldots, T-1$, where T is the total number of frames, the distance between f_i and f_j is defined as follows.

$$d(f_i, f_j) = \sum_{k=0}^{Q_h-1} \left| H_i^h(k) - H_j^h(k) \right| + \sum_{k=0}^{Q_s-1} \left| H_i^s(k) - H_j^s(k) \right| + \sum_{k=0}^{Q_v-1} \left| H_i^v(k) - H_j^v(k) \right| + \gamma \cdot |i-j| \quad (14)$$

The different contribution of the temporal feature can be implemented by adjusting the value of γ. A too small γ will lead to a set of reduced key frames, which merges the similar key frame for different time interval and lose some semantic clue. A large γ will generate a set of elaborate key frames. In addition, since the PCP is a mode-seeking algorithm, it can extract any size of key frame set as one specified according to the typicality order, which is much more flexible.

In the feature space of video sequence, the temporal axis feature is discrete. The PCP algorithm is more suitable for this situation than the PCM algorithm. In addition, the initialization problem is a drawback of the PCM and PCP algorithms. However, in the application of video abstraction, since the video frames are time ordered, we can initialize the PCP algorithm by uniform distributing the clustering prototype patterns along with the temporal axis.

Experimental Results: To illustrate the effectiveness of the proposed PCP algorithm and its application in video abstraction, we conduct two preliminary experiments with synthetic data sets and real news and movie videos. In the first experiment, we create two sets of Gaussian distributed 1000 random patterns in 2D plane with centers at (0,0) and (6,2) and the variance of 1. The FCM, PCM and PCP algorithms are performed on the synthetic pattern set respectively to compare the misclassification numbers, the location bias between the real class centers and the obtained cluster prototypes and the executed CPU time. Here, the location bias is measured by the average Euclidean distance between the class centers and the cluster prototypes. Furthermore, we add 100 noise patterns in one side of the test pattern set, and then compare the classification performance of the three algorithms.

Table 1 shows the performance of the FCM, PCM and PCP algorithms. All the three algorithms are executed in Matlab 5.3 environment with the Pentium III 1.07GHz Mobile CPU and 112M memories. And the parameters in algorithms are specified as m=1.5, c=2. All the data in the Table 1 are the average values of 100 independent experimental results. It is found that for the pattern set without noise, the three algorithms have the similar performance. While for the case of noise contaminated data set, the performance of the PCP and PCM algorithms is better than that of the FCM algorithm. That is to say, the PCP and PCM algorithms are not sensitive to the interference of noise patterns. In addition, the PCP algorithm shows a little advantage over the PCM algorithm in the performance of classification and prototype-extraction. In the experiment, the FCM algorithm is initialized by randomly generating clustering prototypes. While the PCM and PCP algorithms are initialized with the obtained clustering prototypes of the FCM algorithm. Thereby, the FCM algorithm costs less CPU time than the PCM and PCP algorithms. Meanwhile, from Table 1, it can also be observed that the PCP algorithm converges faster that the PCM algorithm, which is agree with the above theoretical analysis.

We have evaluated the performance of the proposed PCP-based key frame extraction algorithm by subjective user studies. Since it lacks of benchmarking or ground truth results for key frame extraction algorithms, so far, we do not yet

Table 1: The performance comparison of the FCM, PCM and PCP algorithms.

Data Set Items	Misclassification Number	Raw Data		Noise Contaminated Data		
		Center Bias	CPU time	Misclassification Number	Center Bias	CPU Time (s)
FCM	10.3	0.12267	0.0386	25.7	0.66596	0.0425
PCM	5.9	0.17447	0.1160	7.9	0.16588	0.1278
PCP	6.6	0.12554	0.0872	7.2	0.15251	0.0962

perform any objective evaluation for the proposed algorithm. Like (Tianming Liu, Hongjiang Zhang, and Feihu Qi, 2003), we select 20 testers to give scores based on their satisfaction to how well the key frames capture the salient content of a clip. The following three-level scales for rating the satisfaction are used: 3. Good, 2. Acceptable, 1. Bad.

The test data library is composed of about 0.5-h news video clips and 0.5-h movie video clips. To compare the proposed key frame extraction algorithm with the common used shot-based method, we first segment the test video into video shots. Then the first frame of each shot is extracted as its key frame. With the obtained total number of video shots, the PCP-based algorithm is employed to extract the corresponding key frames. The evaluation results are shown in Table 2.

Table 2: The evaluation results of the PCP-based key frame extraction algorithm.

Video Type	Algorithms	Good	Acceptable	Bad	Total	Key-frame percentage 0.34%
New Video	Shot-based Method	101 (65.6%)	42 (27.3%)	11 (7.1%)	154	0.34%
	PCP-based Method	124 (80.5%)	28 (18.2%)	2 (1.3%)		
Movie Video	Shot based Method	213 (53.1%)	127 (31.7%)	61 (15.2%)	401	0.89%
	PCP-based Method	306 (76.3%)	85 (21.2%)	10 (2.5%)		

Each row of the table shows the results for a different video type with two methods. The column of total shows the total number of key frames, which corresponds to the number of obtained video shots. The column of Good, Acceptable, and Bad are the numbers of shots that testers give the rate Good, Acceptable, and Bad, respectively. We can see from Table 2 that the performance of the proposed PCP-based method is better that that of the shot-based method. It implies that the first frame of each shot sometimes cannot represent the main semantic features of this shot, especially for the video shots with gradual transition, such as fade-in/fade-out, wipe and dissolve. The percentages over total test clips are also shown in these columns. The column of key-frame percentage stands for the percentage of key frames over total video frames. It is obvious that the percentages of key frames in movie video is several times higher than those of news video, and the video shot values of movie video are also larger than those of news video. This matches the fact that there is usually much more camera shot transitions in movie video than in news video. Moreover, there are much more gradual transition between video shots in movie than those in news video. We can also see that the percentage of Bad rate is very low and the percentage of Good rate is fairly high for the proposed method. This means that the number of and where to place to key frames in our algorithm can capture the salient visual content within a video sequence. In addition, from the compared result, we can draw a conclusion that the proposed algorithm is more suitable for the news video than for the movie video.

Conclusion

In this paper, we have presented a novel possiblistic *c*-patterns clustering with some advantages and wider applications over the famous FCM and PCM algorithms. The application of the proposed PCP algorithm in video abstraction shows its effectiveness. In the one hand, the PCP algorithm can select the representative video frames as key frames; In the other hand, it can extract any number of typical frames as one need or wish to form video abstract. Moreover, it finds the *c* clusters and typical pattern according to the order of the representative or typicality degree. To verify the performance of the PCP algorithm in the video abstraction with various video types and large test video database will be the following work.

Acknowledgement

This work was supported partially by the National Natural Science Foundation of China under grant No.60202004, and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, China.

References

Xinbo, Gao, Weixin Xie, 2000. Advances in theory and applications of fuzzy clustering, Chinese Science Bulletin, 45: 961-970.

Zadeh, L. A., 1965. Fuzzy sets, Inf. and Cont., 8: 338-353.

Dubois, D. and H. Prade, 1988. Possibility Theory: An Approach to Computerized Processing of Uncertainty, New York: Plenum Press.

Pawlak, Z., 1982. Rough sets, Int. J. of Computer and Information Sciences, 11: 341-356.

Ruspini, E. H., 1969. A new approach to clustering, Inf. and Cont., 15: 22-32.

Bezdek, J. C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithm, New York: Plenum Press.

Krishnapuram, R., J. M. Killer, 1996. The possibilistic c-means algorithm: insights and recommendation, IEEE Trans. on Fuzzy Sets, 4: 385-396.

Aigrain, P., H. J. Zhang and D. Petkovie, 1996. Content-based representation and retrieval of visual media: A state-of-the-art review, Multimedia Tools Applicat., 3: 179-202.

Gunsel, B. and A. M. Teklap, 1998. Content-based video abstraction, Proceedings of IEEE Int. Conf. on Image Processing, Chicago, IL, pp: 128-132.

Arthur Pope, Rakesh Kumar, Harpreet Sawhney and Charles Wan, 1998. Video abstraction: Summarizing video content for retrieval and visualization, Proceedings of Thirty-Second Asilomar Conference on Signal, Systems and Computers, pp: 915-919.

Hanjalic, A. and H. J. Zhang, 1999. An integrated scheme for automatic video abstraction based on unsupervised cluster-validity analysis, IEEE Trans. on Circuits and Systems for Video Technology, 9: 1280-1289.

Tianming, Liu, Hongjiang Zhang and Feihu Qi, 2003. A novel video key-frame-extraction algorithm based on perceived motion energy model, IEEE Trans. on Circuits and Systems for Video Technology, 13: 1006-1013.