In Proc. Int. Conf. Pattern Recognition, ICPR'04, Cambridge, UK, August 2004.

Learning classes for video interpretation with a robust parallel clustering method

Vincent Samson and Patrick Bouthemy
IRISA/INRIA,
Campus universitaire de Beaulieu,
35042 Rennes cedex, France
{Vincent.Samson,Patrick.Bouthemy}@irisa.fr

Abstract

We propose an original learning approach for image classification problems. Recognizing semantic events in video requires to preliminary learn the different classes of events. This first stage is crucial since it conditions the further classification results. In video content analysis, the task is especially difficult due to the high intra-class variability and to noisy measurements. We then represent each class by the centers of several sub-classes (or clusters) thanks to a robust partitional clustering algorithm which can be applied in parallel to a (non-predefined) number of classes. Our clustering technique overcome three main limitations of standard K-means methods: sensitivity to initialization, choice of the number of clusters and influence of outliers. Moreover, it can process the training data in an incremental way. Experimental results on sports videos are reported.

1. Introduction

This paper is concerned with the issue of recognizing semantic events in videos and focuses on the learning stage which is of tremendous importance to be able to achieve semantic video interpretation from numerical video features. Such a problem arises in numerous applications such as video summarization, video retrieval or surveillance. In the context of video interpretation, a major concern is the high video appearance variability that a given event may exhibit. Therefore, we have to deal with heterogeneous classes while classes may be not so distant from each other.

Among the various classification techniques, Support Vector Machines (SVM) have become popular since [2]. It is an efficient alternative approach to Bayesian classifiers when no parametric probabilistic distributions are available to model the classes. It is also of interest when dealing with numerous and correlated features. SVM methods directly seek an optimal separating hyperplane in the feature space, usually on a two-class problem basis. However, in practice, an appropriate trade-off has to be found between the fidelity to training data and the non-linearity of the decision function. This may become a tricky task when dealing with a complex class topology as illustrated in Fig. 1. Furthermore, the presence of noise and possibly outliers is inherent to real data.

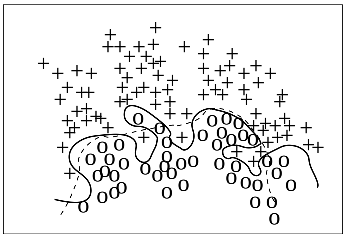

Figure 1. Example of 2D data corresponding to two different classes exhibiting a high intra-variability, and candidate SVM nonlinear decision functions (dashed and solid lines).

As mentioned above, the detection of semantic events in videos usually involves heterogeneous classes. For a given class, observations, and consequently computed video features, may vary according to the way the scene is filmed (camera motion, distance to the scene, illumination conditions) and the considered instance of the event of interest. This is the case for example in sports video analysis where a class of a given "play" event is actually reflected by several clusters in the feature space of low-level motion descriptors (depending on the camera parameters and on the athlets being filmed). SVM do not take into account this intra-class variability. Moreover, they are often limited to two-class classification, since multi-class extension is not so straightforward and still an active subject of research [7].

We suggest instead to consider a robust clustering approach applied in parallel to each predefined class in order to capture their internal data structure. Such an approach is intrinsically multi-class. It is flexible (classes can be handled in any order) and extensible (new classes can be straightforwardly added). It is also incremental with respect to training data (new sets of training data can be processed just by iterating from the current learning state). The training data consist of a set of vectors collecting the extracted features on each video sample of the training set. For the sake of simplicity, all attributes are supposed to be numerical and continuous.

The learning approach advocated in this paper involves a novel intra-class clustering algorithm that can be seen as a robust extension of the standard K-means algorithm. The algorithm yields an estimated number of clusters for each class and associated prototypes (actually the centers of the subclasses), from which different strategies of classification could be considered.

The reminder of the paper is organized as follows. Section 2 describes related work on robust and so-called unsupervised clustering. In Section 3, we first introduce the general formulation of K-means-type clustering methods. Our contribution is then motivated and explained in detail. Section 4 is devoted to the experimental results, and Section 5 contains concluding remarks.

2. Related work

Clustering techniques are classically divided into two broad categories: hierarchical and partitional algorithms [8]. Among the partitional methods, the K-means algorithm is perhaps the most widely used. Originally devised as an *online* clustering technique, most people refer to it as a *batch* algorithm. It provides a "hard" partition of the data as opposed to its "fuzzy" counterpart called fuzzy C-means. The common purpose of center-based clustering algorithms is to summarize multivariate data by a reduced set of central points. It is very close to vector quantization design which consists in encoding a signal source with codebook reference vectors, as in the well-known LBG algorithm [11]. We briefly review hereafter partitional approaches to both *robust* clustering in the presence of noisy data and *unsupervised* clustering when the number of clusters K is unknown.

The general technique to handling the problem of K-means sensitivity to outliers is to modify the objective function by considering an additional noise cluster or by incorporating concepts from robust statistics. The K-medoids algorithm is one of the earliest solution but most algorithms have been derived in the framework of fuzzy clustering so that it can deal with overlapping cluster boundaries [3]. Note also that the notion of robust vector quantization developed in the context of data encoding and transmission does not refer to outliers but to channel noise and to random elimination of prototypes for codebook reduction [6].

The traditional method to determining K is to select the K-means partition that optimizes a certain validity measure over a range of K values [1]. Nevertheless, the computational cost is quite high and the choice of appropriate validity indices evaluating the quality of the partition still remains a difficult question [5]. One alternative is to perform some progressive clustering by starting with an over-specified number of clusters and then adding a second phase of merging. "Stepwise" clustering and "dynamic" local search have also been suggested [9] but all these solutions need to specify similarity criteria and to set thresholds.

Very few techniques attempt to globally solve the unsupervised robust clustering problem. In [4], the authors propose a fuzzy clustering algorithm involving a robust "competitive agglomeration" process. A regularized objective function is put forward with the use of fuzzy memberships and a robust loss function, while a separate virtual noise cluster is introduced in [10] to catch the outliers. Even so the number of clusters can be derived in that way, such a regularization approach requires in practice several heuristics to tune the parameters of the considered criterion.

3. Intra-class robust clustering

3.1. K-means clustering

Let $\mathcal{X} = \{\mathbf{x}_n\}_{n=1,\dots,N}$ be a set of feature vectors (in our case, they are the training data from a same given event class). Each data instance contains d real-valued attributes: $\mathbf{x}_n = [x_{n1}, \dots, x_{nd}]^t \in \mathbb{R}^d$.

K-means clustering consists in finding K clusters such that a global distortion error is minimized. The standard objective function is actually a sum of variances within each group. It involves the K centers of each cluster (often called prototypes) and the criterion amounts to the minimization of the sum of distances to the nearest prototypes:

$$E(\mathcal{Y}_K) = \sum_{n=1}^{N} \min_{l \in \{1,\dots,K\}} \mathcal{D}(\mathbf{x}_n, \mathbf{y}_l) \qquad (1)$$

where $\mathcal{Y}_K = (\mathbf{y}_1, \dots, \mathbf{y}_K)$ is the unknown K-tuple of prototypes. Conventional K-means algorithm uses the square Euclidean distance: $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2 = \sum_{i=1}^{d} (x_i - y_i)^2$, but other center-based clustering algorithms can be derived by using different measures of distortion or by assigning a weight to each data point.

Even in the standard case, minimizing the objective function requires an iterative procedure and getting the global minimum is not guaranteed. Optimization starts with some given prototypes, and alternates data allocation to the nearest prototypes and update of the prototypes. Such iterative techniques only converge to a local minimum of the objective function. Moreover, the number K of clusters must be known *a priori*.

3.2. Description of our robust clustering method

First of all, we replace the square Euclidean distance traditionnally used, by a robust distance so that the influence of outliers on prototype estimation will be brought down. We have chosen $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \rho_c(\|\mathbf{x} - \mathbf{y}\|)$ where ρ_c is the hard redescending Tukey's biweight defined as follows:

$$\rho_c(u) = \begin{cases} u^6/6c^4 - u^4/2c^2 + u^2/2 & \text{if } |u| < c, \\ c^2/6 & \text{otherwise.} \end{cases} \qquad (2)$$

This function is quadratic at the origin, increasing for small positive values and constant after a threshold c. Thus, points which distance to a prototype is greater than c have no influence on its determination. They are considered as outliers for this given cluster. Secondly, we do not fix *a priori* the number of clusters K, but only the scale parameter c of the robust distance. The algorithm starts with a single cluster, and progressively creates new clusters from the data considered as outliers with respect to the existing clusters. K and c are obviously related: the number of estimated clusters will grow if c decreases. It is yet a consistent advantage not to fix K, since a rough estimate of c can be derived from statistics computed on the training data and the obtained partition is far less sensitive to a change in c than in K. Moreover, we can automatically and easily determine K by choosing the optimal value of c for which the partition is the most stable (i.e., the number of clusters and center locations remain unchanged when iterating). Thirdly, we can set an appropriate initialization of cluster centers by searching for dense regions in the feature space of the training set. We extract local density information from statistics precomputed on data samples, by evaluating the number of neighbor points of these samples for a given neighborhood size (determined from the histogram of pairwise distances).

The overall learning scheme consists in partitioning for each predefined class its associated training set with the robust clustering algorithm described as follows:

1. normalize the input data (so that each attribute has a median absolute deviation equal to one and therefore a similar influence on the distance measure);

2. compute pairwise distances between vectors (for a subset of randomly selected vectors if necessary), let $c = c^{(1)}$ be the median of pairwise distances and initialize the first prototype at the center of the most dense region;

3. iterate the following steps

   - allocate each data point to the nearest prototype,
   - label the non-allocated data as outliers and possibly create a new cluster if there exists a sufficiently dense region of outliers (that contains more than a predetermined number of points),
   - reestimate the locations of the prototypes,

   until convergence of the prototypes;

4. decrease the value of c ($c^{(q+1)} = c^{(q)} - \Delta c$, index q being the iteration number) and repeat step 3, the prototypes being initialized with the previously estimated prototypes;

5. stop when the partition is stable ($K^{(q+1)} = K^{(q)}$ and $\sum_l \|\hat{\mathbf{y}}_l^{(q+1)} - \hat{\mathbf{y}}_l^{(q)}\|^2 / \sum_l \|\hat{\mathbf{y}}_l^{(q)}\|^2 < \epsilon$);

6. rescale the prototypes to the initial vector space.

Note that we begin by a preliminary normalization of the different features. As a consequence, we can use an isotropic distance metric $\rho_c(\|\mathbf{x} - \mathbf{y}\|)$ with a single scale parameter c. It must also be mentioned that center updating corresponds to a nonconvex minimization procedure that we solve by using iteratively reweighted least squares (IRLS) together with a continuation method similar to the "Graduate Non Convexity" algorithm.

An incremental version of our algorithm can easily be implemented to take into account supplementary data: start with the current prototypes and include the new data of the given class to reestimate their locations and perhaps create one or several new cluster(s) in each class.

4. Experimental results

4.1. Description of data sets

We have used in our experiments two distinct datasets extracted from two groups of videos, one related to amateur basket-ball games and the other to tennis TV programs. We are interested in detection and classification of video events based on the analysis of the dynamic content. We use thereby motion descriptors extracted from the normal flow magnitudes of each video sequence and their temporal contrasts, and corresponding to the empirical parameters of a Dirac-Gaussian mixture modeling their histograms [12].

The first set, *Basket-ball*, is composed of only 2D feature vectors so as to easily illustrate the behavior of our robust clustering algorithm (the two parameters of the Dirac-Gaussian mixture representing the 1D histogram of the temporal contrasts only). All of the sequences are short videos of one or two basket-ball players filmed by an amateur with a static camera. They involve different players and variable shooting conditions (Fig. 2). We have defined 3 classes of semantic events named Middle shot, Lay-up and One-on-one. We report in the next section the clustering results obtained on the training set. The second dataset, called *Tennis*, consists of the entire 4D motion descriptors characterizing video segments of a tennis TV program and the goal is to recognize 4 categories of events in the video (Rally, Serve, Change of side and No play segments). In subsection 4.3, we also evaluate our approach in terms of classification results. For that purpose, each dataset is divided in a training set and a test set.

4.2. Learning results

Fig. 3 displays the clustering results obtained on the *Basket-ball* training dataset composed of 189 feature vectors unequally divided into the 3 classes. For each class, our unsupervised algorithm selects a different number of clusters which correctly structure the processed data.

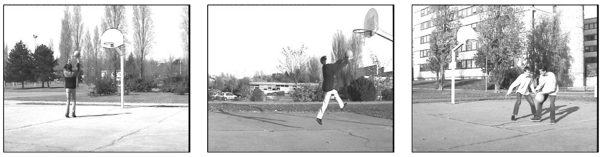

Figure 2. Three image samples extracted from the *Basket-ball* video dataset and corresponding to the 3 different types of actions: “middle shot”, “lay-up” and “one-on-one”.

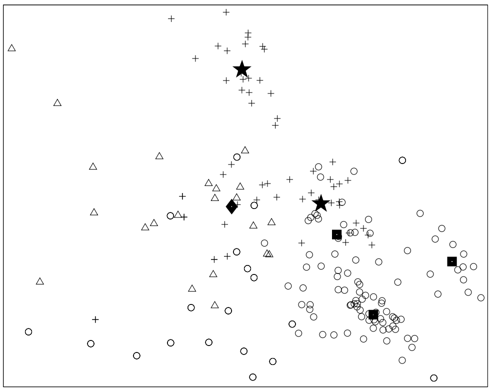

Figure 3. *Basket-ball* dataset: the training points are represented by different symbols according to their class (‘o’ for “middle shot”, ‘+’ for “lay-up” and ‘Δ’ for “one-on-one”). We have obtained partitions of respectively three, two and one clusters. The corresponding centers are marked with dark squares, pentagrams and a diamond respectively.

We have also applied our learning algorithm on the 4-class *Tennis* training dataset with 300 data points. We obtain two clusters for Rally, four clusters for Serve, three for Change of side and two for No play. Associated parameter $c^\star$ varies between 0.4 and 2.2 depending on the dispersion of the data in each class.

4.3. Classification results

Our learning method handles each class separately but, for classification purpose, we need to specify a proper decision rule based on the computed prototypes of each class. The most simple one is probably the minimum distance classifier. Our intra-class clustering algorithm allows us to determine several prototypes per class that can be further used to classify data. Here we resort to the nearest prototype rule. This basic classifier gives a better classification rate on the *Basket-ball* test set compared to the “one-against-one” LIBSVM classifier [7] with a Gaussian kernel and hyper-parameters tuned by cross-validation (87% against 80% out of 123 examples). The results are equivalent on the *Tennis* dataset (67% of good classifications out of 339 test segments), while our learning method presents significant advantages: it is straightforwardly extensible to new classes, it is incremental by construction if new training data are available, and it does not require any parameter selection beforehand.

5. Conclusion

We have presented a novel partitional robust clustering algorithm for learning the structure of each class in video content analysis. Our approach is flexible, parallel and unsupervised. It can determine the appropriate number of prototypes in a well formalized and simple way. It is also incremental and robust to outliers, contrary to Gaussian mixture model clustering for instance. Experimental results show the capacity of our method to parsimoniously and usefully representing classes of video events. Future work will deal with the design of a prototype-based classifier that could improve the simple nearest neighbor strategy.

Acknowledgments: This work was supported by the European IST project LAVA: Learning for Adaptable Visual Assistants (IST-2001-34405).

References

[1] J. C. Bezdek. Some new indexes of cluster validity. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28(3):301–315, June 1998.

[2] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[3] R. N. Davé and R. Krishnapuram. Robust clustering methods: A unified view. *IEEE Trans. on Fuzzy Systems*, 5(2):270–293, 1997.

[4] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Trans. on PAMI*, 21(5):450–465, May 1999.

[5] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, December 2001.

[6] T. Hofmann and J. M. Buhmann. Competitive learning algorithms for robust vector quantization. *IEEE Trans. on Signal Processing*, 46(6):1665–1675, June 1998.

[7] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Trans. on Neural Networks*, 13(2):415–425, March 2002.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surv.*, 31(3):264–323, Sept. 1999.

[9] I. Kärkkäinen and P. Fränti. Dynamic local search for clustering with unknown number of clusters. In *ICPR’02*, Quebec, August 2002.

[10] B. Le Saux and N. Boujemaa. Unsupervised robust clustering for image database categorization. In *ICPR’02*, Quebec, August 2002.

[11] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Comm.*, 28:84–95, 1980.

[12] G. Piriou, P. Bouthemy, and J.-F. Yao. Learned probabilistic image motion models for event detection in videos. In *ICPR’04*, Cambridge, UK, August 2004.