Revisiting the K-means Algorithm for Fast Trajectory Segmentation

Luis A. Leiva* Enrique Vidal*
Institut Tecnològic d'Informàtica, Universitat Politècnica de València

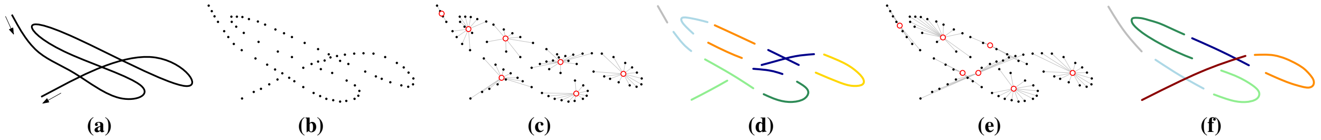

Figure 1: *A 2D example. An arbitrary trajectory (1a) is digitized at a constant sampling rate (1b), and our objective is to club the path coordinates together in, say, 7 clusters. K-means based algorithms do not deal with temporal information, therefore groups are ill-defined (1c) and hence also the resulting segmentation (1d). Our proposal, though, allows to easily cope with the sequentiality of data (1e,1f).*

Keywords: sequential data, trajectories, clustering, segmentation

1 Introduction

Many problems in Computer Science require a trajectory segmentation, in part due to the notably huge spectrum of devices that capture sequentially-generated information (e.g., motion sensors, video cameras, RFID tags, eye trackers, etc.) Segmentation leads to simplify the structure of the data, so that original objects can be divided into smaller, more compact structures. Seen this way, segmentation can be approached as a compression technique, i.e., organizing trajectories into segments whose members are similar in some way. This can be solved as a clustering problem. Unfortunately, to date we have not found a suitable method that can tap in a really simple way the temporal constraint implicitly embedded in the data. Moreover, near-optimal solutions such as kernel methods or hidden Markov models can be prohibitive if processing power is a restriction (e.g., on mobile devices).

Here we present a novel trajectory segmentation technique based on the K-means algorithm, a special case of EM clustering. K-means is well known for its simplicity, relative robustness, and really fast convergence to local minima. It is also well known that its performance depends upon two key points: initial partition and instance order. For that reason, we tuned the algorithm of [Duda et al. 2001] for unsupervised classification. This version, instead of using the classical minimum distance criterion [Lloyd 1982], is a sequential, iterative optimization refinement that evaluates the sum-of-squared error (SSE, also denoted as J_e) at each step, reallocating a sample to a different cluster if and only if that reassignment decreases J_e. Also, given the sequentiality of data, we use the trace segmentation (TS) algorithm [Kuhn et al. 1981] for centroid initialization. TS consists in a non-linear sampling operation that redistributes objects to enforce even spacing between them, eliminating thus redundancy.

2 Method

We use TS to build an initial partition of the data. Then we impose the following constraint to the classification step of K-means: a sample $\boldsymbol{x}$ in segment i is iteratively reallocated to the previous or next clusters ($i \pm 1$), characterized by their means $\boldsymbol{\mu}$ and their number of samples n. The best reassignment j^* is determined if the variation in SSE is beneficial, i.e., when $\Delta J_e < 0$:

$$j^* = \underset{i-1 \le j \le i+1}{\arg\min} \; \Delta J(i,j)$$

where

$$\Delta J(i,j) = \frac{n_j}{n_j+1} \parallel \boldsymbol{x} - \boldsymbol{\mu}_j \parallel^2 - \frac{n_i}{n_i-1} \parallel \boldsymbol{x} - \boldsymbol{\mu}_i \parallel^2 .$$

The algorithm stops when there are no samples to reallocate, ensuring thus that the partition has reached the minimum error boundary.

3 Contributions and Benefits

Our proposal: *1) is accurate*, since it guarantees the convergence to the "best" local minimum, i.e., the less distorted segmentation of the original trajectory; *2) is robust*, as each run for a given K always yields in the same segments — thanks to the TS initialization; *3) is fast*, because, instead of the classical one-against-all strategy of search, we only need to check two clusters in each classification step — computational complexity is thus $\Theta(kd)$ instead of $\Theta(nkd)$; *4) does not require extra input parameters*, just the sample vectors and the number of desired segments, as in K-means; *5) is specially suited for real-time applications and large datasets*, since the computational cost of updating the centroids is independent of the number of samples; and *6) supports on-line learning*: clusters can be updated while new objects arrive without affecting the previous data structure.

4 Conclusion

Ours is a really straightforward method to simplify the segmentation of motion based trajectories. The introduced modifications to (the sequential version of) K-means allowed us to deploy a succinct clustering algorithm for segmentation while greatly accelerating the process to convergence. We believe this work opens a new door to novel applications in the computer graphics domain and beyond.

Acknowledgements. Work supported by the research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

Interactive Demonstrator. Please visit the following URL:
`http://personales.upv.es/luileito/wkm/siggraph-demo.html`

References

Duda, R. O., Hart, P. E., and Stork, D. G. 2001. *Pattern Classification*, 2nd ed. John Wiley & Sons, ch. Unsupervised Learning and Clustering, 517–599.

Kuhn, M. H., Tomaschewski, H., and Ney, H. 1981. Fast nonlinear time alignment for isolated word recognition. In *Proc. ICASSP*, 736–740.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Trans. on Information Theory 28*, 2, 129–137.

*e-mail: {luileito,evidal}@iti.upv.es