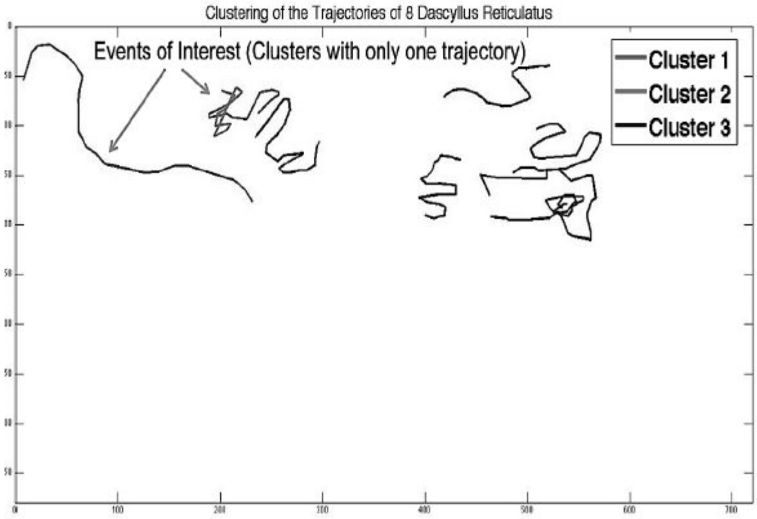
“Automatic Fish Classification for Underwater Species Behavior Understanding” Resume

Motivation/Introduction Ideas

- Goal: classify automatically fishes in the natural underwater environment and detect anomalous behaviors.
- In the past, and in some cases nowadays, fishes are analysed by literally human underwater observation. This is considered invasive and it probably won’t capture normal fish behaviors.
- In the fishing context, vessels use sound techniques (SONAR) in order to detect other boats and fishes in the area.
- One solution would be to install cameras in the underwater environment, it wouldn’t be so invasive. On the other hand, this would create one underlying problem: the amount of data that is gathered daily is too much for being manually analysed by marine biologists. It also would require a robust storage system in order to efficiently store that huge quantity of information.
- Manually analysis (by marine biologists) is considered tedious and also error prone for the same reason.
- In order to detect anomalous behaviors, it’s necessary to have a detection module (to detect fishes in a given frame), a tracking module (to build a movement track of a certain fish) and, in some cases, also a classification module (to distinguish among different species).

Fish Trajectory Analysis System

- First, trajectories pass through a preprocessing module where the Douglass Peucker algorithm is applied. This algorithm reduces the number of points of each trajectory keeping the similarity with the original.
- The preprocessed trajectories are separated in different sets, depending on its species. Then, a clustering algorithm is applied on each set. The anomalous behaviors will be identified analysing produced clusters.
- It’s considered events of interest all the trajectories that belong to a cluster that has few samples, when compared to the total number of samples for that species.
- IKmeans was the chosen clustering algorithm.



(c)

Douglass Peucker Algorithm

- Idea: recursively divide the original line segment, deciding if the farthest point is to be kept or not based on the distance
- Initially the starting and ending points are automatically marked as “to keep”

- Algorithm:

1) Mark first and last point to be kept
2) Get the farthest point to the line segment
3) if distance $\leq \varepsilon$: the points in between are excluded

   if distance > ε: the farthest point is kept; apply first and second steps to the line segments starting point, farthest point and farthest point, ending point

IKmeans Algorithm

- Clustering tasks can be difficult in the time series context due to its properties: high dimensionality and high feature correlation.
- Classic KMeans has two drawbacks: the need of choosing the number of clusters and the highly dependence of centroids initialization. However, KMeans is a lot of times the chosen algorithm due to its fast running time.
- The proposed algorithm takes advantage of the wavelet decomposition (more specifically Haar wavelet decomposition) in order to represent a time series in different resolution levels. This technique allows to find a representation at a lower dimensionality preserving the original information and the original shape.
- Haar wavelet decomposition is achieved by averaging adjacent values, to form a smooth lower dimensional signal. The coefficients are important for reconstructing the original time series.

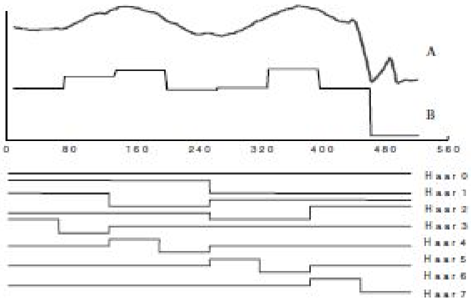


Fig. 1. The Haar Wavelet representation can be visualized as an attempt to approximate a time series with a linear combination of basis functions. In this case, time series A is transformed to B by Haar wavelet decomposition, and the dimensionality is reduced from 512 to 8.

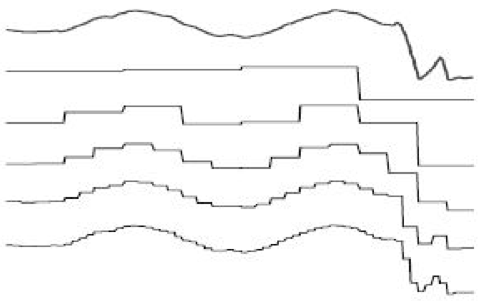
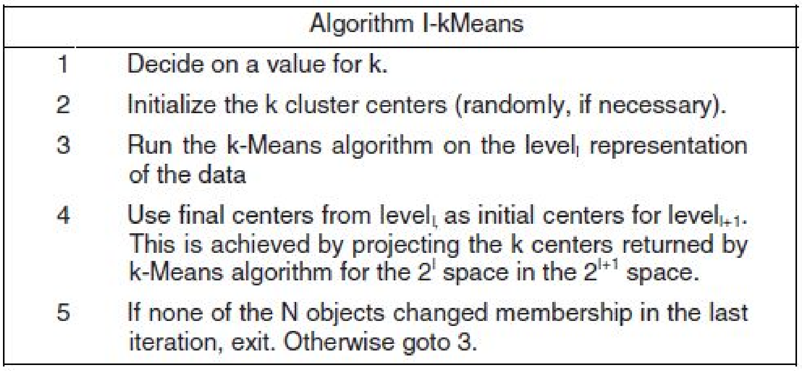


Fig. 2. The Haar Wavelet can represent data at different levels of resolution. Above we see a raw time series, with increasing faithful wavelet approximations below.

- IKmeans idea: apply kmeans at increasingly finer levels of resolution. The resulting centroids of the last level are used in the next one. This is considered an anytime algorithm in a way that the user can interrupt and get results of the last level before starting processing the next one.

Table 2. An outline of the I-kMeans algorithm



	Algorithm I-kMeans
1	Decide on a value for k.
2	Initialize the k cluster centers (randomly, if necessary).
3	Run the k-Means algorithm on the level$_l$ representation of the data
4	Use final centers from level$_l$ as initial centers for level$_{l+1}$. This is achieved by projecting the k centers returned by k-Means algorithm for the 2^l space in the 2^{l+1} space.
5	If none of the N objects changed membership in the last iteration, exit. Otherwise goto 3.

- The algorithm starts on the second level (2^{i-1} dimensions) where i is the level.
- Intuition: the general shape of a time series can be approximately captured at a lower resolution.
- Notes: typically stabilize at a low level thus less processing time needed when comparing to run literally on raw data; when iterating to the next level, the centroids will be better and consequently the number of iterations will be lower.
- The centroid values, when passed to the next level, have to be duplicated in order to match its dimensionality (0.5, 1.2) → (0.5, 0.5, 1.2, 1.2).

Results

- IKmeans algorithm gave better results in terms of cluster quality and also performance (even if it reaches a high level of resolution).

Doubts relative to the approach

- Which distance metric is used? DTW?
- How does the centroid is calculated in that case?

Proposed improvements

- Integrate species events (e.g. predador fish chasing small fish) due to the fact the anomalous behavior can be correlated with other species behavior.