UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di laurea in Fisica

Autore: Marco Zanchi
matr. 885359

A Machine Learning Approach to Breaststroke

Relatore: Prof. Stefano Carrazza

Correlatore: Prof. Stefano Forte

Anno Accademico 2018-2019

Abstract

The aim of this work has been to study and analyse the breaststroke swimming style of several swimmers with methods of machine learning, a tool which has shown great potentials in the field of sports, but it has not been tested yet in examples of breaststroke.
The main purpose has been to find a method for distinguishing different types of swimmers and for classifying them. Furthermore, this work has been focused to find the best way to compare performances of two different swimmers or different performances of the same swimmer. This method could be helpful to train the swimmer to achieve a particular result and improve his swimming style.
To reach the goal, the swimmers' speed time dataset provided by the Dipartimento di Scienze Sportive, Università degli studi di Milano, has been analysed, and the best swim of each athlete has been isolated. With best swim it is meant the single succession of stroke and flutter kick where the swimmer gains more acceleration. The best swim has been used to compare different swimming performances of different swimmers to find the main features of the breaststroke style. Then, the affinity propagation algorithm has been applied to four different similarity matrices, constructed from the best swim profiles with four different metrics: the euclidean distance, the mean squared error, the mean absolute error and the Kolmogorov-Smirnoff test.
This approach has verified that the affinity propagation algorithm is able to recognise different performances of the same swimmer, clustering them in the same class. Furthermore, the comparison of the different performances of the same swimmer or of several swimmers, has confirmed the importance of the flutter kick in attaining high breaststroke speed. Finally, based on the results of this work, three kinds of specific trainings can be proposed which aim to improve the swimming style and

the performances of the athletes.

Contents

Chapter 1

Introduction

1.1 Aim and forward of the work

Machine learning is currently being applied as an analytical and problem solving tool in a wide variety of fields from theoretical physics, to autonomous driving, online suggestions, fraud interception, and to instantaneous language translation. Machine learning analysis can be also used in sport fields to improve athletes performance and help them to determine the best way of training [1]. The advantage is that machine learning is able to quickly analyse complex datasets and multiple data sources.

There are very few studies that have tried to use machine learning algorithms to determine training planning on high levels, especially in swimming sports [2]. The overall goal of the present study is to illustrate the potential of machine learning in sports on poorly investigated examples of breaststroke. In particular, this research focuses on the classification of swimmers' performance using the affinity propagation algorithm for the improvement of their skills.

The dataset on which that machine learning algorithm has been applied has been provided by Dipartimento di Scienze Sportive, Università degli Studi di Milano and it consists of speed time data of 66 breaststroke swimmers.

1.2 Breaststroke

The breaststroke is one of the four main swimming styles with the front stroke, the backstrokes and the butterfly stroke and it is the slowest one among them [3]. In the breaststroke, the swimmer swims with the chest down in the water, arms slightly breaking the surface of the water, legs always underwater and the head underwater for the second half of the stroke.

The kick is sometimes referred to as "frog kick", because of the resemblance to the movement of a frog's hind legs. However, when done correctly, it is more similar to a "whip kick". The body is often at a steep angle to the forward movement, which slows down the swimmer more than any other style.

A swimming event can be composed in four moments or phases: the starting phase, the swimming phase, the turning phase and the finishing phase [4]. Since the swimming phase, composed in turn by an active and passive pahse, is the most important one, it will be deeply analysed in the successive paragraphs. A scheme of the movements succession is shown in 1.1.

The active phase of the stroke - that gives propulsion - is deep and varied. Deep because the hands search the water in depth and varied because the arms vary their position with respect to the trunk in relation to the sensitivity of the swimmer on the water.The pushing phase does not take place as the hands are never set back with respect to the head.

The retropulsion of the upper limbs coincides, in the technical swim, with the adduction of the shoulder blades to the column that provides greater force on the water. This gesture, in connection with the posterior arch of the spine and the pressure of the underlying water, allows the swimmer to break the surface from the head to the lumbar area. The arms are then recovered outside or near the surface of the water, but in any case the elbows must remain submerged and close to the body.

In the passive phase, the swimmer tries to exploit the height reached with respect to the water to go as far forward as possible by placing the

shoulder blades and flexing the column. Consequently, at the end of stroke recovery, the gluteal surface emerges from the surface, reducing friction with the fluid. In this swim, the recovery of the legs (abduction and flexion) begins when the traction ends (end of the active stroke). This movement is immediately followed by a simultaneous adduction and extension of the legs (active phase). The action of the leg is often characterized by a low excursion on the lateral plane of the legs. Moreover, in this sort of scissor kick, the flexing foot gradually adjust itself following the direction of the leg. It follows that the foot will return "hammer" only at the end of the recovery, thus also reducing friction in the passive phase.

The effectiveness of the leg also depends on the mobility of the swimmer's joints, in fact greater mobility allows the swimmer to push more water and therefore to go faster. In breaststroke, a breathing happens at each stroke cycle, but the head should remain in line with the torso [5]. The performance of the breaststroke style strongly depends on the coordination between the upper and lower limp [6] [7] Breaststroke speed depends 70% on the action of the legs, while the remaining 30% is due to the action of the arms.

1.3 Machine learning

Machine learning can be considered a type of applied statistics which, as defined by Goodfellow [8], increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions. Using a machine leaning algorithm implies that the algorithm is able to learn from data. To understand what is meant by learning from data, it is possible to quote Mitchell [9]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T, as measured by P, improves with experience E." . Machine learning can solve several tasks, as classification,

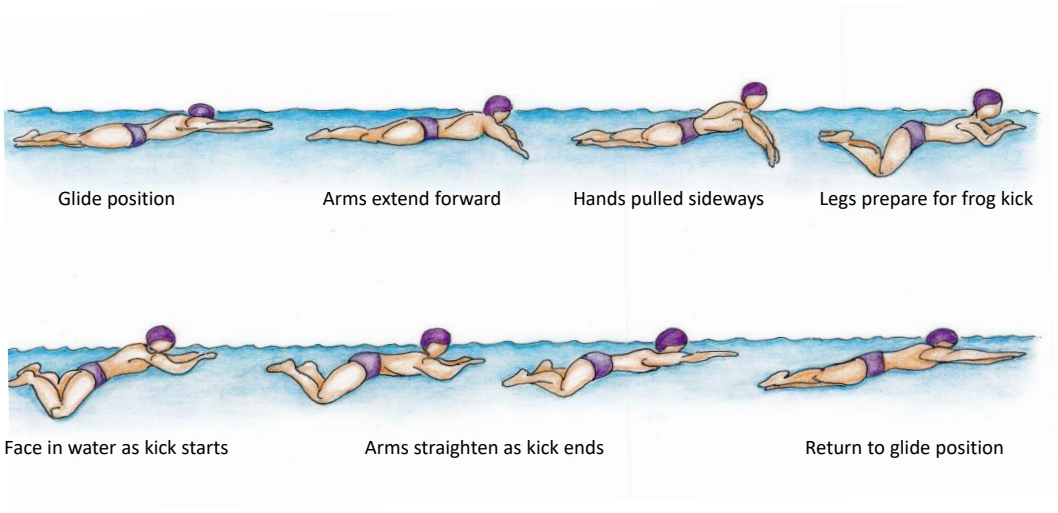

Figure 1.1: Breaststroke style scheme

classification with missing inputs, regression, transcription, machine translation, synthesis and sampling. To check how a machine learning algorithm learns, it is useful to define a quantitative measure of its performance. For each different task T, a specific measure P should be defined. In case of classification, classification with missing inputs, and transcription, the accuracy of the model should be measured. Accuracy is the percentage of correct outputs produced for a certain number of examples. Another way is to measure the percentage of incorrect outputs.

In the literature, machine learning algorithms are classified as unsupervised or supervised based on the experience they are allowed to have during the learning process. Briefly, unsupervised learning algorithms are applied to dataset containing many unclassified data, and learn information on the data only based on the structure of the dataset. Instead, supervised learning algorithms elaborate information starting from targeted or labelled data.

Machine learning is so important and popular nowadays in science and industry because, it provides a fast way to gain information and build

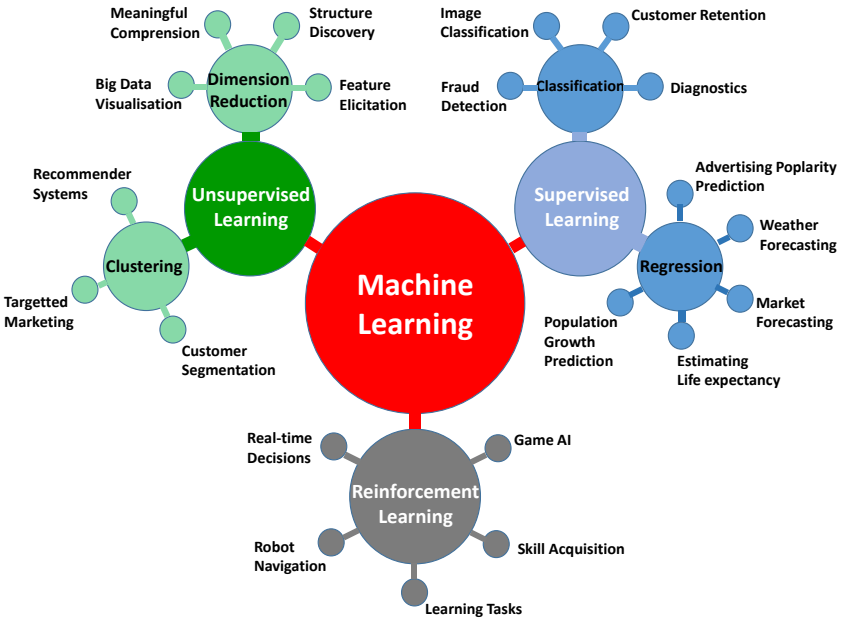

Figure 1.2: Graphical representation of the most popular machine learning applications

models using the data only - and obviously the machine learning algorithms. Those methods of big data analysis have spread fast in different fields, as theoretical physics, autonomous driving, online suggestions, fraud interception, and instantaneous language translation (a scheme of that interdisciplinary applications can be observe in 1.2).

Machine learning ideas have developed enormously in the last 70 years, following the development of artificial intelligence, and computer science. The main statistical tools became available before the 1940 with the Bayes theorem (1812), the Least Squares method for data fitting (1805) and the Markov Chains (1913). These methods are very important to machine learning. The first step of machine learning development could be dated to the late years of 1940s with the invention of stored-program computers. Those computers held their programs in the same memory used for data. The most important ones were the Manchester Small-Scale Experimental Machine in 1948, Cambridge's EDSAC and the Manchester Mark 1 in 1949, and the University of Pennsylvania's EDVAC in 1951. In 1950, Alan

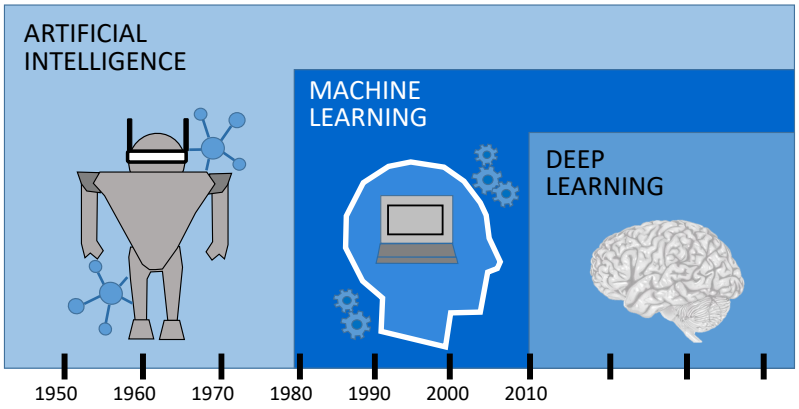

Figure 1.3: Machine learning history

Touring published Computing Machinery and Intelligence (REF) in which he wonders if computers could be defined intelligent. In 1951, Marvin Minsky and Dean Edmonds (REF) that simulated the work of organic brains created the first neural network. After this fast evolution, artificial intelligence suffered of a period characterized by many failures REF), until the late 1980s, when new tools as informatics, machine learning and computational intelligence made their appearance. The next step was the invention of deep learning, a subsystem of machine learning, which is based on the use of neural networks (REF). A scheme of machine learning development can be observed in 1.3.

In the successive paragraph it is shown how machine learning operates. Machine learning takes as input a certain type of data (the form of this data depends on the specific task) then the algorithm creates models or clusters analysing the structure of the data which could be labeled or not. Different types of machine learning applications have been developed according to the enormous amount of uses, which that subject could be applied. There are two principal branches of machine learning :

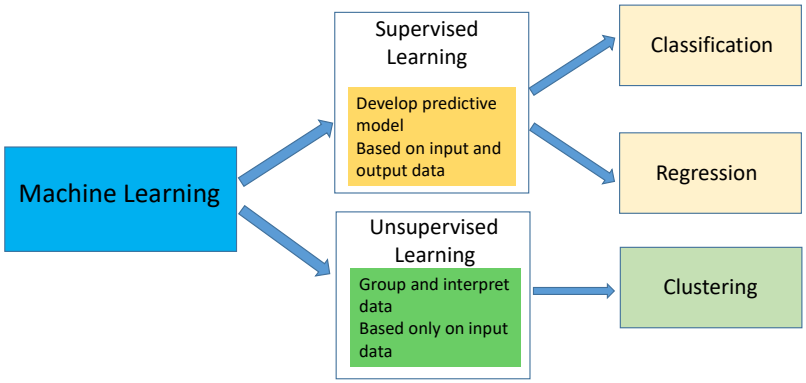

Figure 1.4: An example of most common machine learning branches

the supervised learning and the unsupervised learning (as it has been schemed in 1.4). The supervised learning algorithm takes as input labelled data, for example, creating the model using the information provided from the supervisor. Data are often grouped as the input and output, training the algorithm to create a model, which can be applied to new input data in order to forecast a new output. The main algorithm is the k-nearest neighbour algorithm. The unsupervised learning takes as input unlabelled data and creates a model analysing the data structure. One of the most important application of unsupervised learning is cluster analysis. The aim of this application is to classify and group unlabelled data identifying similarities among the data. The main algorithms are the affinity propagation and the k-means. Another important branch of machine learning is the reinforcement learning. This technique consists in the construction of a system of reward and punishment, which trains the algorithm to learn a 'decisional path'.

Chapter 2

Analysing breaststroke signals

2.1 Filtering

The signal analysed is a periodic speed time function which represents the variation of the swimmer' s speed in the time (thet is named swimmer' s profile). An example of the signal shape is shown in 2.1. One of the most important tools used in the present work is the filtering of the entire swimmers' profiles obtained by plotting each dataset in speed time graph. The aim of this tool is to filter and cut the best swim (the single repetition of stroke and flutter kick).

For each profile, the best swim is considered the one characterized

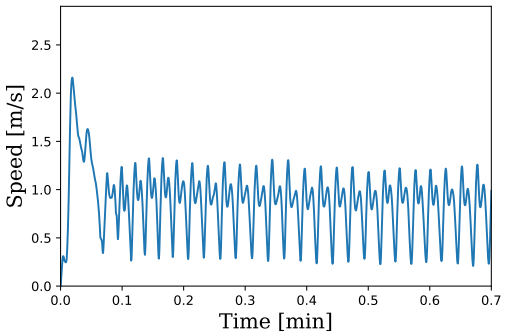

Figure 2.1: Example of an entire swimming profile, that is the shape of the variation of the swimmer' s (y-axis) in the time (x-axis).

by the period where the swimmer gains more acceleration. To do so, the speed gained between the start of the stroke and the peak of the flutter kick (corresponding to the maximum extension of the flutter kick) has been divided by the time spent on. The single movement which has the maximum acceleration is the best period.

The first problem to deal with is how to isolate that period. In a first attempt, a Fourier transform signal analysis has been performed to isolate the periodic part cutting the irregular parts. Unfortunately, due to fluctuations in the speed measuring, the algorithm was not able to recognise the periodic shape. Another method to filter the swimmer's profile has been to apply the change of the sign of derivative as a counter to isolate periods.

A program has been built with Python in order to take as input the swimmer's speed time dataset and return the part of the dataset which represents the best period. The count of the sign of derivative changes has been built taking into account several restraints to check for fluctuations that could distort the count of sign changes. The algorithm takes as input the csv file containing the speed time dataset and, using two different functions, it returns the best period. The first function filters the profiles excluding the non periodic part (i.e., cutting the fluctuations). Then, it creates a vector which stores the maximum and minimum of the speed of the profiles with their position in the original dataset (that allows to reconstruct the best profile with its original speed and time). The second function takes as input the output of the first function and it uses the counter of changing derivative sign computed cutting the extremes of the best period. To do this, the algorithm computes for each period the acceleration as the difference between the first minimum and the second maximum speed (gained from the start of the stroke to the maximum extension of the flutter kick) divided by the time spent, and it chooses as best period the one which has the largest acceleration. Then, the algorithm, using those extremes, recreates the original best period and returns it as the output.

2.2 Metrics for the accessment of swim profile

After the best period has been filtered, four metrics have been introduce to compare the different swimmers' profiles. The metrics selected are the euclidean distance, the mean squared error[10], the mean absolute error [11], the Kullback-Leibler divergence [12] the Kolmogorov-Smirnoff test [13]. Those metrics have been also used to build the similarity matrix for the affinity propagation algorithm. It has been decided to use different types of metrics to study and compare the different results produced by the affinity propagation algorithm and to understand which of those metrics is the best one for that work. In those subsections all the metrics used are briefly introduced.

2.2.1 Mean squared error

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of a procedure for estimating an unobserved quantity measures the average squared difference between the estimated value and the actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{2.1}$$

2.2.2 Mean absolute error

In statistics, mean absolute error (MAE) is the measure of the difference between two continuous variables. Assuming that X and Y are variables expressing the same phenomenon, it is possible to interpret the mean absolute error as follows: considering a scatter plot of n points, where point i has coordinates (xi, yi), the Mean Absolute Error (MAE) is the average vertical distance between each point and the identity. MAE is

also the average horizontal distance between each point and the identity line. The mean absolute error is given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - y_i\right| \tag{2.2}$$

2.2.3 Kullback Leibler divergence

In mathematical statistics, the Kullback–Leibler divergence is a measure of how one distribution is different from a second one. In the simplest case, a Kullback–Leibler divergence of 0 indicates that the two distributions considered are identical.

$$D_{KL}(P||Q) = \sum_{x\in\chi} P(x)\log\frac{P(x)}{Q(x)} \tag{2.3}$$

2.2.4 Kolmogorov Smirnoff test

In statistics, the Kolmogorov–Smirnov test is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution, or to compare two samples. The Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples.

2.3 Best period reconstruction

The profiles obtained by the best period filtering have different dimensions. Indeed, they consist of a different number of time-speed couples. Since the metrics used in the analysis take as input two vectors of the

same dimensions (containing the speed of the best period profile), it has been necessary to remake the best period profiles with all the same dimensions. A way to achieve this, is to use an interpolate method with Python language, in this case the spline method, to obtain all the profiles with the same dimensions(the function `make_interp_spline` from scipy.interpolate has been used [14]).

Another method used has been to add zeros at the beginning of the speed vector to obtain equal dimensions. In the present work, the results of the analysis based on the spline method are reported in an exhaustive manner, whereas the results of the second method are only briefly described, due to the low quality of the obtained graphs.

2.4 Affinity Propagation

The machine learning tool used in the present work is the algorithm named affinity propagation. Affinity propagation is a clustering machine learning algorithm, that clusters data based on a measure of similarity. It has been published by Frey et al. in 2007. It is a method that simultaneously considers all data points as potential exemplars [15]. It is possible to understand how it works by considering each data point as a node in a network where real-valued messages are transmitted along its edges until a good set of exemplars and corresponding clusters have been founded. The exemplars are the data which represent the class. It is possible to refer to them as class centres. Messages among the points are updated according to minimize an appropriately chosen energy function. The method is called "affinity propagation" because the value of each message reflects the affinity of one data point with another point, that can be chosen as its exemplar [15].

Affinity propagation takes as input an ensemble of real values which are the similarities between data points. The similarity $s(i,k)$, where i and k are two possible objects of the input collection, shows how well

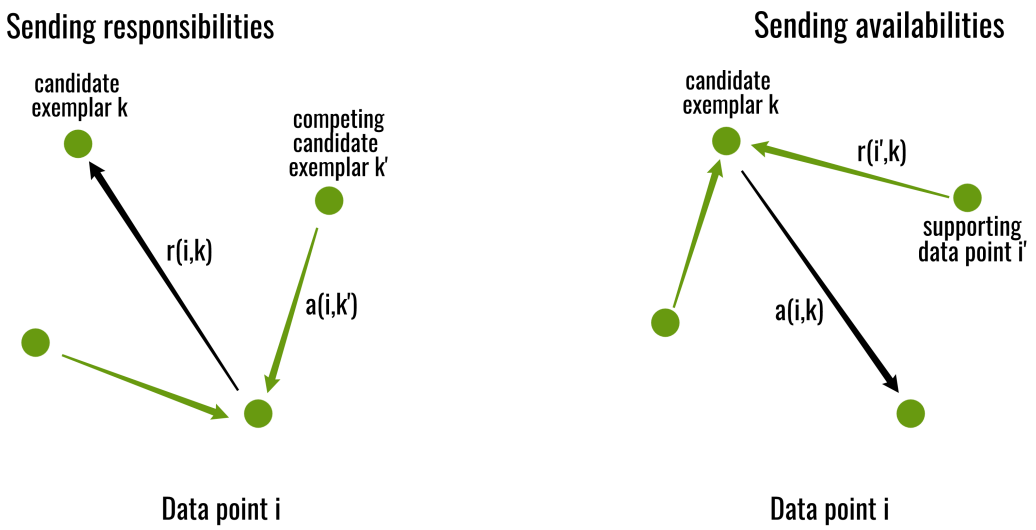

Figure 2.2: An example of responsibility and availability message passing [15]

the object k is an acceptable exemplar for object i. It is relevant to observe that data points with larger values of $s(k,k)$ are more suited to be selected as exemplars. Those values are referred to as "preferences." The affinity propagation algorithm takes as input also a parameter called preference, which represents the preferences described above. That parameter influences the number of identified exemplars, which is also the number of clusters. The number of clusters also depends on the message-passing procedure.

Responsibility and availability are two kinds of messages exchanged between data points which consider a different kind of competition, deciding which points are exemplars and to which centre the other points belong. According to Frey et al. [15] responsibility and availability are defined as follows The "responsibility" $r(i,k)$, sent from data point i to candidate exemplar point k, represents how acceptable point k is to be the exemplar for point i, considering other potential exemplars for point i. The "availability" $a(i,k)$, sent from candidate exemplar point k to point i, represents how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points

that point k should be an exemplar. The operation of availability and responsibility massage passing is illustrated in 2.2 and shown in the successive paragraphs. The availabilities are initialized to zero: $a(i,k) = 0$. Then, the responsibilities are computed using the rule

$$r(i,k) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \{a(i,k') + s(i,k')\}. \tag{2.4}$$

In the first iteration, due to the fact the availabilities are set to zero, the $r(i,k)$ value is the initial similarity between point i and point k, considered its exemplar, minus the largest of the similarities between point i and other potential exemplars. This initial computation does not consider how many other points prefer each candidate exemplar. In later iterations, when some points are assigned to other exemplars, their availabilities value will become negative, decreasing the real values of some of the input similarities $s(i,k')$ in (2.5), thus removing that candidate exemplars from the competition.

For $k = i$, the responsibility $r(k,k)$ is set to the input preference that point k be chosen as an exemplar, $s(k,k)$, minus the largest of the similarities between point i and all other candidate exemplars [15]. This "self-responsibility" reflects the evidence, accumulated in the previous interactions, that point k could be an exemplar. That evidence is based on its input preference which is modified by how it does not adapt to be clustered with another exemplar. Even if the responsibility computed above makes all the candidate exemplars compete for gaining a data point, the following availability update considers how data points select each candidate exemplar as a good one for themselves

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r(i',k)\}\}. \tag{2.5}$$

The availability $a(i,k)$ is computed as the self-responsibility $r(k,k)$ plus

the sum of the (positive) responsibilities that the candidate exemplar k receives from other points [15]. Only the positive portions of incoming responsibilities are selected to be added in the previous update, because it is only necessary for a good exemplar to explain some data points well (positive responsibilities), regardless of how poorly it explains other data points (negative responsibilities) [15]. The negative value of self-responsibility $r(k,k)$ indicates that point k prefers to belong to another exemplar rather than becoming an exemplar itself. If that situation occurs, the availability of point k to be an exemplar can be increased if there are other points having positive responsibilities for point k being their exemplar [15]. To limit the effect of incoming positive responsibilities, the total sum is limited avoiding it to become negative. The "self-availability" $a(k,k)$ is computed differently

$$a(i,k) \leftarrow \sum_{i's.t.i' \neq k} max\{0, r(i',k)\}. \tag{2.6}$$

This message reflects the evidence, accumulated in previous interactions, that point k is an exemplar. That fact is based on the positive responsibilities sent to candidate exemplar k from other points [15].

Affinity propagation algorithm presents many advantages then other methods, as $k-centres$ clustering. The popular k-centres clustering algorithm takes as input an initial set of randomly selected exemplars of the dataset and iteratively refines this set decreasing the sum of squared errors. K-centres clustering depends on the initial selection of exemplars implying lots of rerun with different initial inputs finding a suitable solution. However, this approach works well only when the number of clusters is small. Affinity propagation has various advantages over other clustering techniques. Methods such as k-centres clustering, k-means clustering, and the expectation maximization algorithm compute only a small set of estimated cluster centres at each step, compare to the affinity propagation. They also strongly depend on the initialization set.

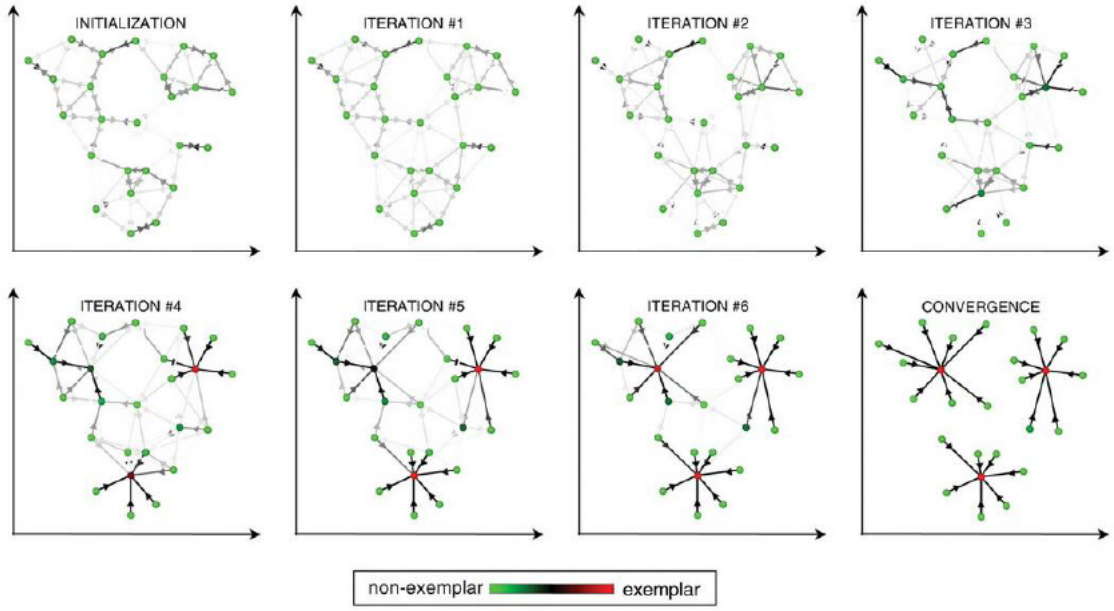

Figure 2.3: An example of affinity propagation creating classes in different iterations [15]

In contrast, by simultaneously considering all data points as potential centres and gradually identifying clusters, affinity propagation is able to avoid many of the poor solutions caused by unlucky initializations and hard decisions [15]. Other methods as Markov chain Monte Carlo randomly search for good solutions, but they do not have the affinity propagation's advantage of considering many possible solutions simultaneously. Affinity propagation finds clusters with a much lower error than the other methods, and it does so in less than one-hundredth the amount of time.

An example of affinity propagation work is shown in 2.3 where affinity propagation is illustrated for two-dimensional data points. The negative Euclidean distance has been used to measure similarity. Each point is colored according to the current evidence that it is a cluster center (exemplar). The darkness of the arrow directed from point i to point k corresponds to the strength of the transmitted message that point i belongs to exemplar point k [15].

Chapter 3

Analysis and results discussion

3.1 Dataset

The dataset consists of 66 different CSV sheets containing the data for 35 swimmers collected by the Dipartimento di Scienze Sportive, Università degli Studi di Milano. Each of these sheets contains the instantaneous speed and position for any instant of time of a performance of one swimmer. The interval of time at which the speed and position have been recorded are irregular. Each of these Excel sheets represents the swimmer's profile. Three swimmers have nine profiles each, recorded during different performances. Two swimmers have three profiles each, recorded while the athletes were carrying an extra weight. All the other swimmers have one profile each.

3.2 Profiles

The first step of the analysis of the breaststroke has been to plot the dataset available for each swimmer, using the matploitlib library of Python. From these speed time graphs, it is possible to observe the periodic motion of stroke and flutter kick (3.1). The graphs show the characteristic shape of the breaststroke style, which consists of a periodic repetition of

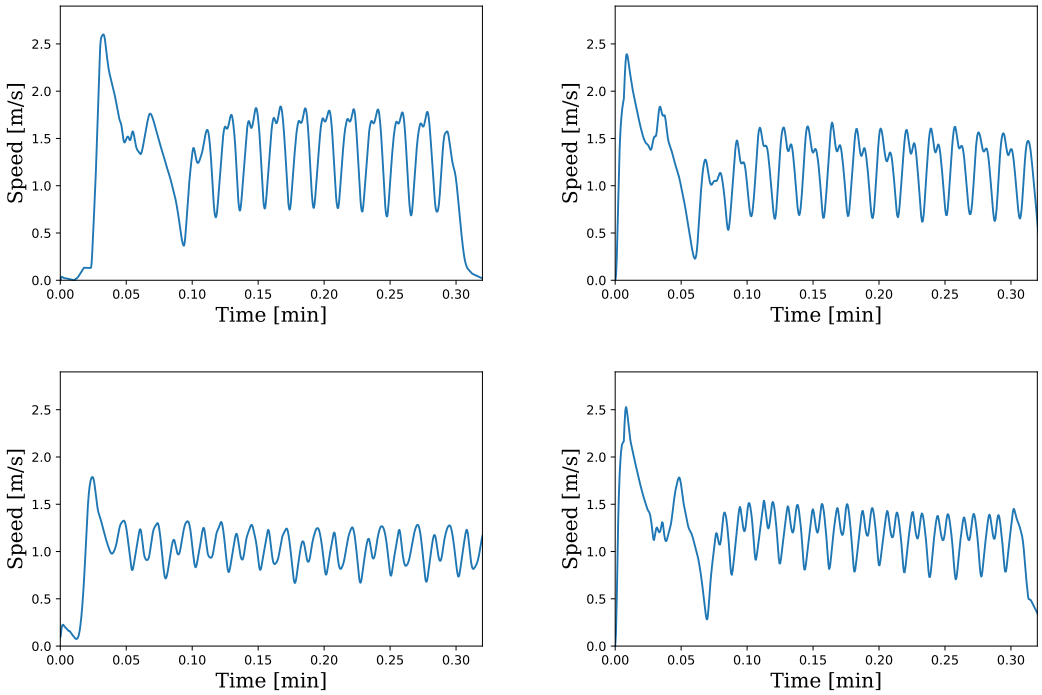

Figure 3.1: Four examples of entire profiles of different swimmers. Notice the periodic shape of the breaststroke style.

two peaks, representing the succession of stroke and flutter kick. Some of the obtained swimmer's profiles are reported in Fig. 3.1.

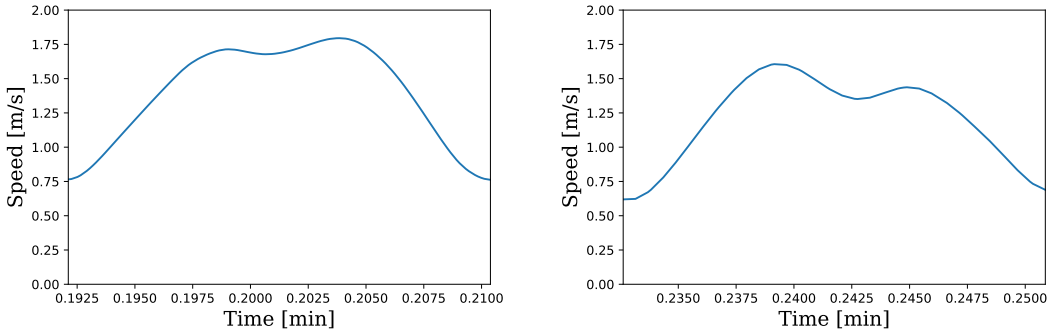

Figure 3.2: Two examples of best period profiles

3.3 Filter

The results of the period filtering described in the tools section are shown below. Analysing the shape of the best period, it is possible to understand the style of the swimmer. For example, in the first plot (3.2, left) the swimmer takes a fast stroke and an even faster flutter kick. However, the second swimmer (3.2, right) takes a fast stroke, but a slower flutter kick.

3.4 Metrics and confront

Once all the swimmers' best periods have been reconstructed, some metrics have been selected to compare and analyse their swimming profiles. Those metrics are the euclidean distance, the mean squared error,the mean absolute error, the Kullback-Leibler divergence and the Kolmogorov-Smirnoff test. The metrics have been collected form the library statistics of sklearn module [16]. The metrics have been linearized to create a linear comparison between two swimmers. The distance between all swimmers has been computed and plotted to find a linear behavior. Unfortunately, a linear behaviour has not been found, as the 3.3 shows.

Another method to calculate a quantitative difference between two swimmers has been to compute the percentage ratio between the average

speed of the two swimmers. This method allows to quantitatively calculate the speed loss between two swimmers or between two different swims of the same swimmer, providing also the opportunity to control the swimmer's style and thus to improve it. In section 3.8 it is possible to see an application of this method.

3.5 Clusters analysis with spline

Once all the best period profiles with equal dimensions obtained by the spline method have been collected, the successive step has been to cluster them to find different classes of swimmers. To perform this, the algorithm of Affinity Propagation has been used, due to the fact that it does not require to know a priori the number of clusters. The algorithm was taken from the library cluster of module sklearn [16]. The algorithm takes as input the matrix of distances between all the best period profiles, so there are different results for each metric used to compute the distance matrix. In particular, only the speed has been selected, and not the time of each swimmer. With this algorithm, it was not possible to use the Kullbac-leibler divergence due to problems of divergence.

In the subsection below, the results of the swimmers' clustering for each metric are shown. First, a plot of the clusters has been produced with the embedding algorithm (named MDS) of the manifold library from the sklearn module [16]. Unfortunately, in that case, the MDS algorithm has not been able to create a manifold where the points are clustered in different groups. Indeed, although the affinity propagation algorithm has been able to create different clusters, the MDS algorithm has not been able to produce a plot that represents an ensemble of clusters. So it has been decided not to include those plots in the present work. Then, it has been supposed that most of the average speeds of the cluster's elements are normally distributed. So the median of the average speeds of the cluster's elements and the standard deviation (computed as the difference between the extreme of the sorted average speed vector taken out the

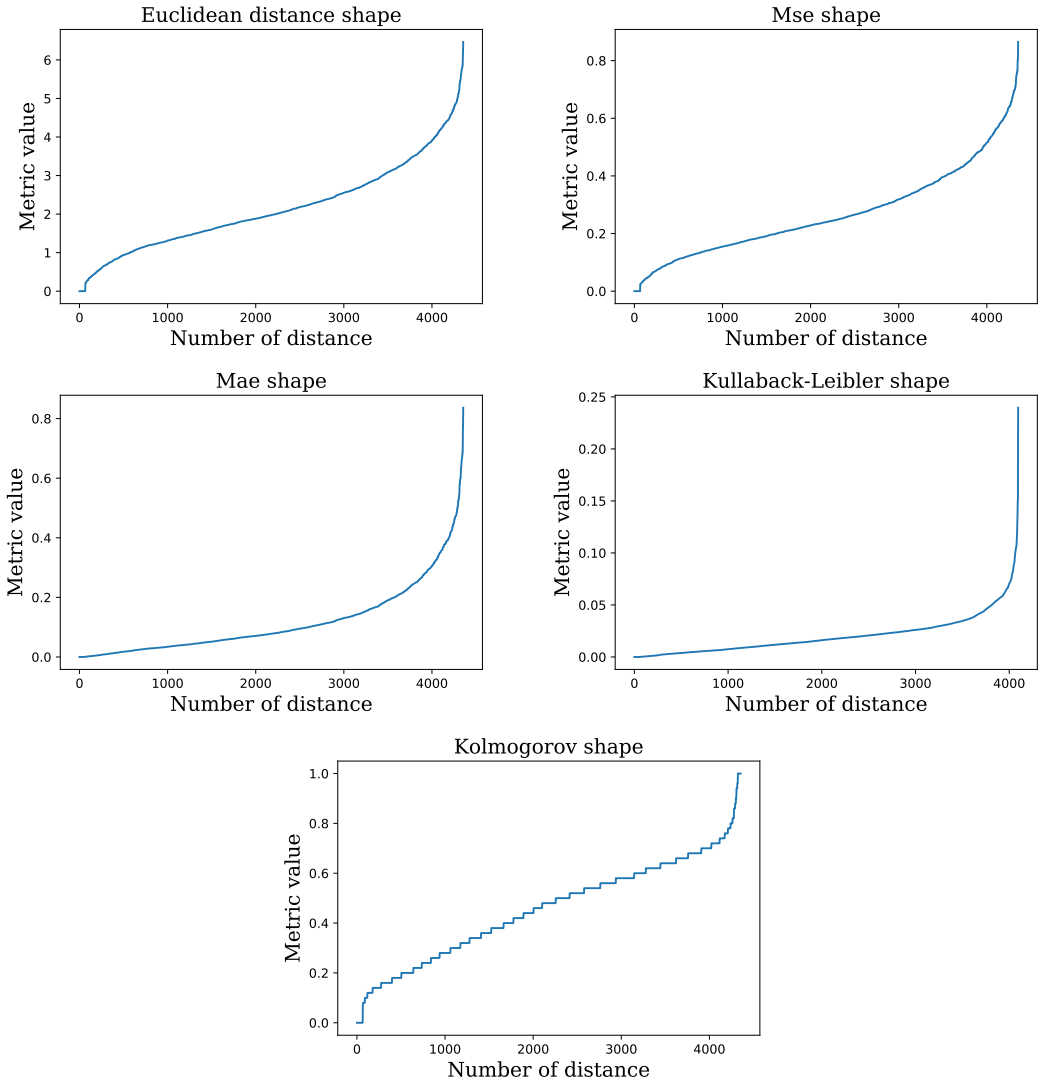

Figure 3.3: Shape of pairwise evaluation of metrics between all the swimmers.

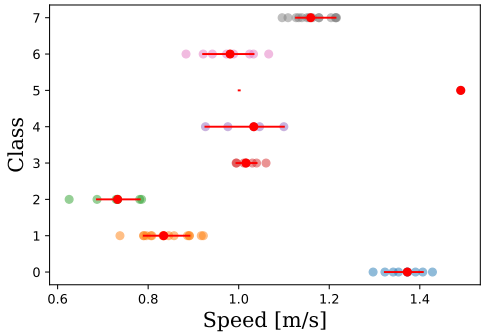

Figure 3.4: Euclidean distance. Left: median speed and std of each cluster. Right: average speed distribution of the elements of each clusters.

15% from the head and the tail) of each cluster have been computed and plotted as point and error bar in a number class speed graph (Red color in 3.4, 3.7, 3.10, 3.13). The distribution of the average speed of the elements of each cluster has been plotted in a speed class graph (Transparent colors in 3.4, 3.7, 3.10, 3.13). The subsequent graphs show the shape of each class in a speed time graph coupled with the ratio plot between each member and the class center (3.5, 3.8, 3.11, 3.14, 3.15). The shape of all the centers has been plotted in a speed time graph and the average speed of each center in a speed number class graph (Last box in 3.6, 3.9, 3.12, 3.16). All the graphs have been produced with the matploitlib library of Python.

3.5.1 Cluster results and analysis with euclidean distance

Using the Euclidean distance to compute the distance matrix, eight clusters have been produced. It is possible to observe the classes sape in Fig. 3.5. Class 0 contains nine elements with similar shape. Class 1 contains 15 elements with different shape. Class 2 contains seven elements with different shape. Class 3 contains seven elements with similar shape. Class 4 contains five elements with similar shape. Class 5 contains only one elements. Indeed, it has an anomalous shape. Class

6 contains eight elements with different shape. Class 7 contains 14 elements with different shape.

3.5.2 Cluster results and analysis with mean squared error

Using the mean squared error to compute the distance matrix, seven clusters have been obtained. It is possible to observe the classes shape in 3.8. Class 0 contains nine elements with similar shape. Class 1 contains 16 elements with non similar shape. Class 2 contains seven elements with non similar shape. Class 3 contains only one elements. Indeed, it has an anomalous shape. Class 4 contains 12 elements with non similar shape. Class 5 contains 14 elements with non similar shape. Class 6 contains seven elements with similar shape.

3.5.3 Cluster results and analysis with mean absolute error

Using the mean absolute error to compute the distance matrix, eight clusters have been produced. It is possible to observe the classes shape in 3.11. Class 0 contains seven elements with non similar shape. Class 1 contains seven elements with similar shape. Class 2 contains six elements with non similar shape . Class 3 contains only one elements. Indeed, it has an anomalous shape. Class 4 contains eight elements with non similar shape. Class 5 contains 14 elements with non similar shape. Class 6 contains nine elements with similar shape. Class 7 contains 14 elements with non similar shape.

3.5.4 Cluster results and analysis with kolmogorov-smirnoff test

Using the Kolmogorov-Smirnoff test to compute the distance matrix, nine clusters have been obtained. It is possible to observe the classes shape

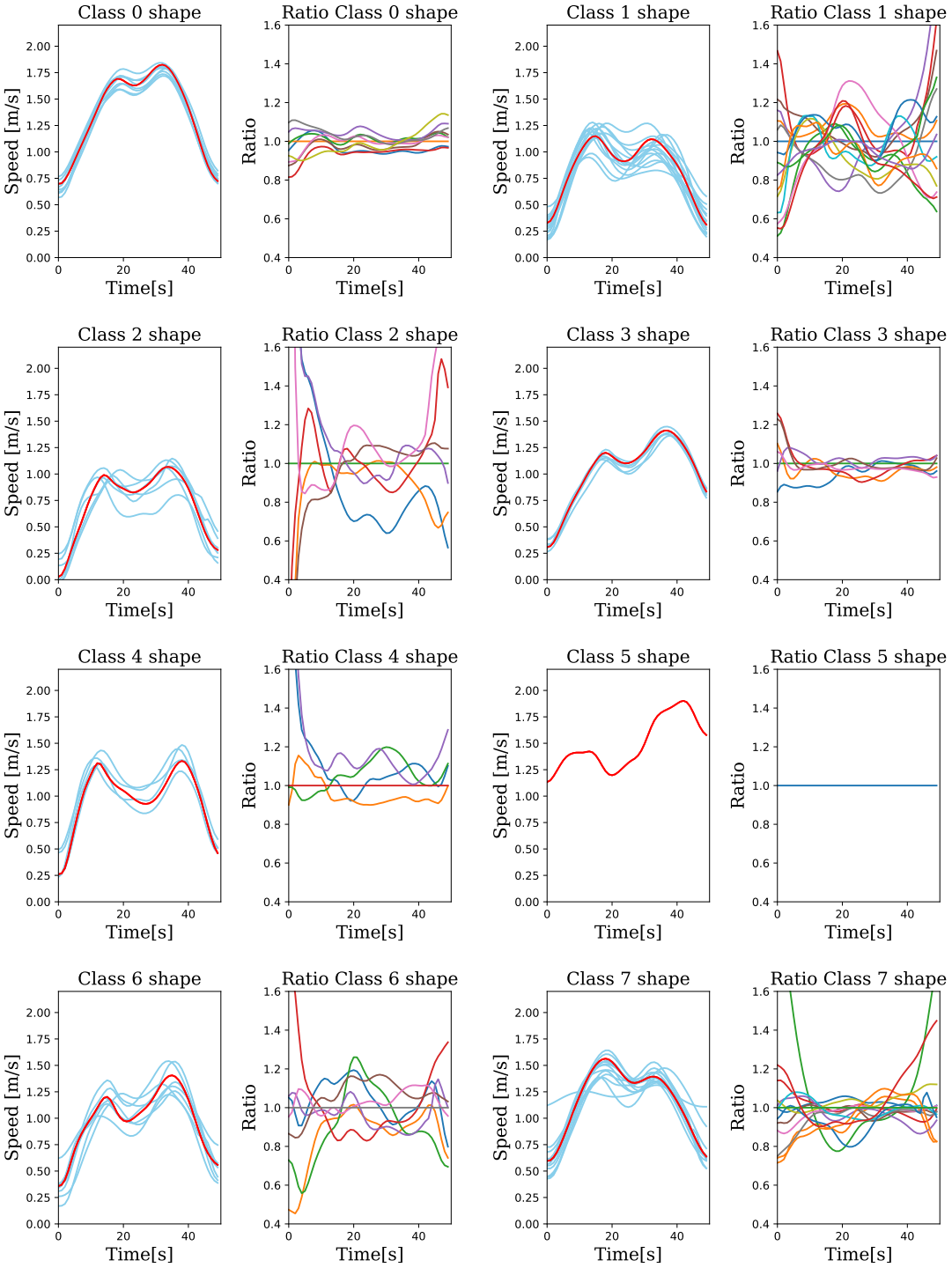

Figure 3.5: Euclidean distance. Plots of class shape (left) and ratio class shape (right)

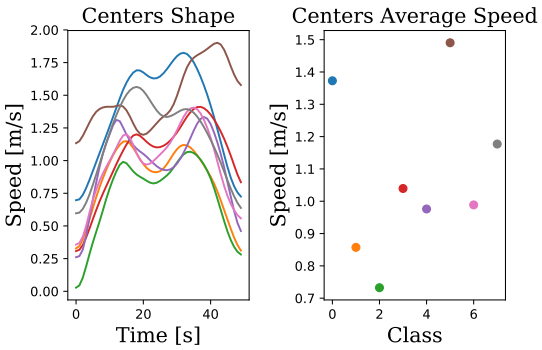

Figure 3.6: Euclidean distance. Plot of centers shape(left) and centers average speed(right).

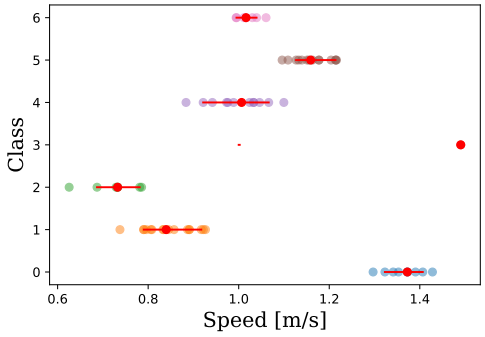

Figure 3.7: Mean squared error. Median speed and std of each cluster (red).Average speed distribution of the elements of each clusters (transparent colors).

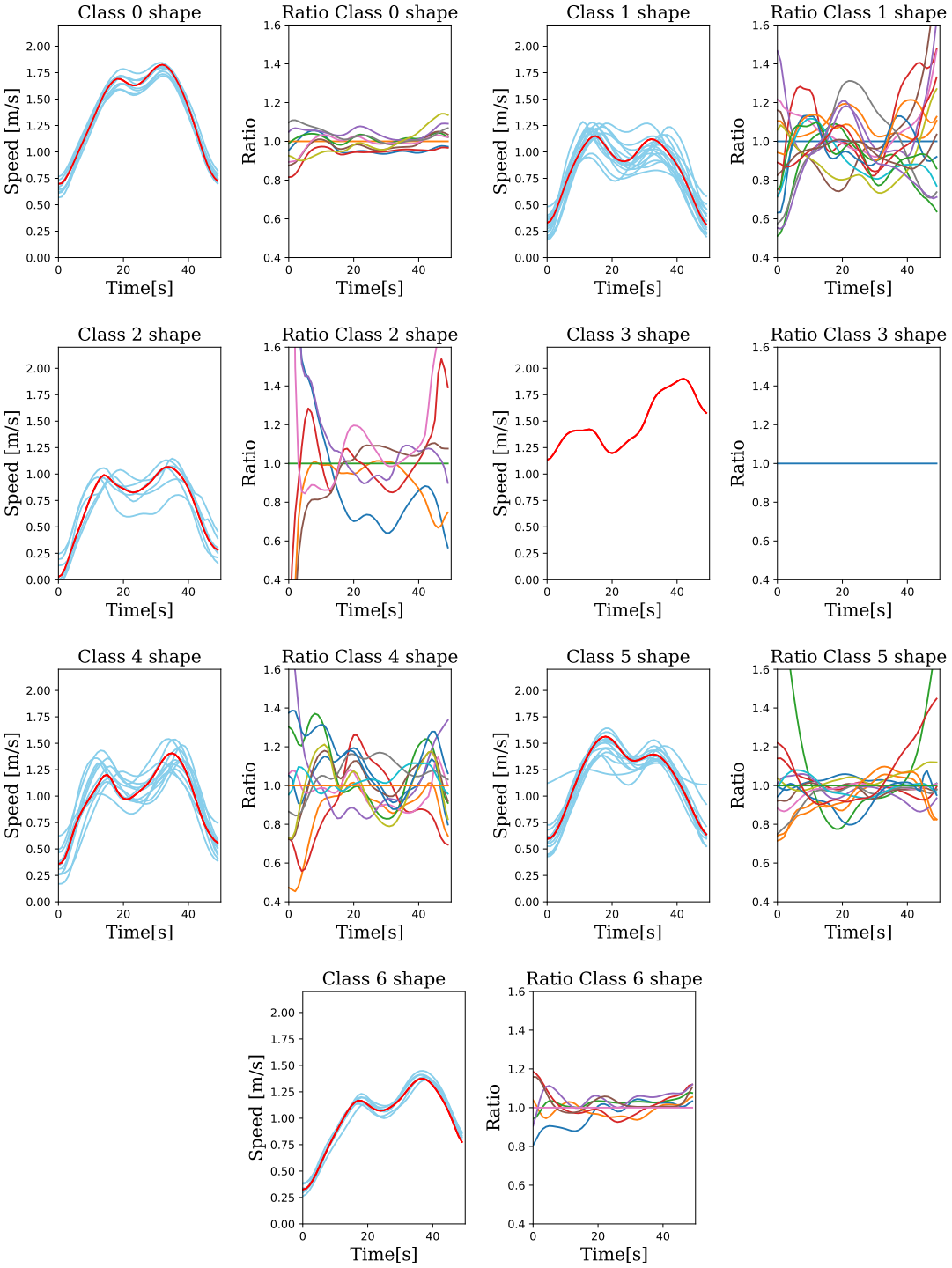

Figure 3.8: Mean squared error. Plots of class shape (left) and ratio class shape (right).

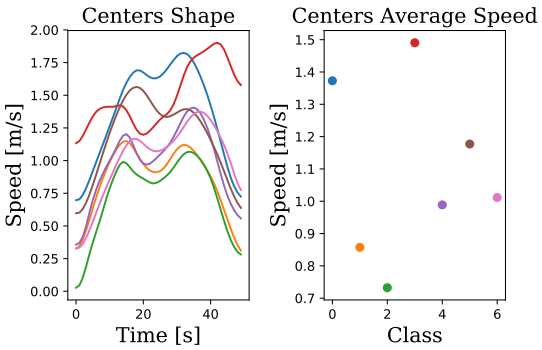

Figure 3.9: Mean squared error.Plot of centers shape (left) and centers average speed (right)

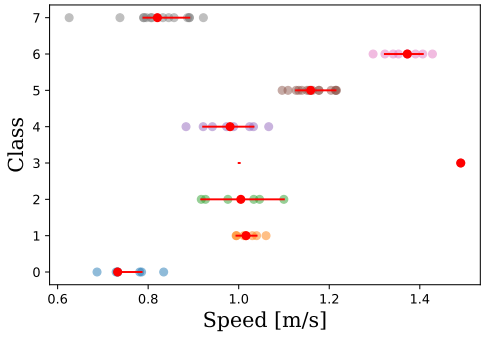

Figure 3.10: Mean absolute error. Median speed and std of each cluster (red).Average speed distribution of the elements of each clusters (transparent colors).

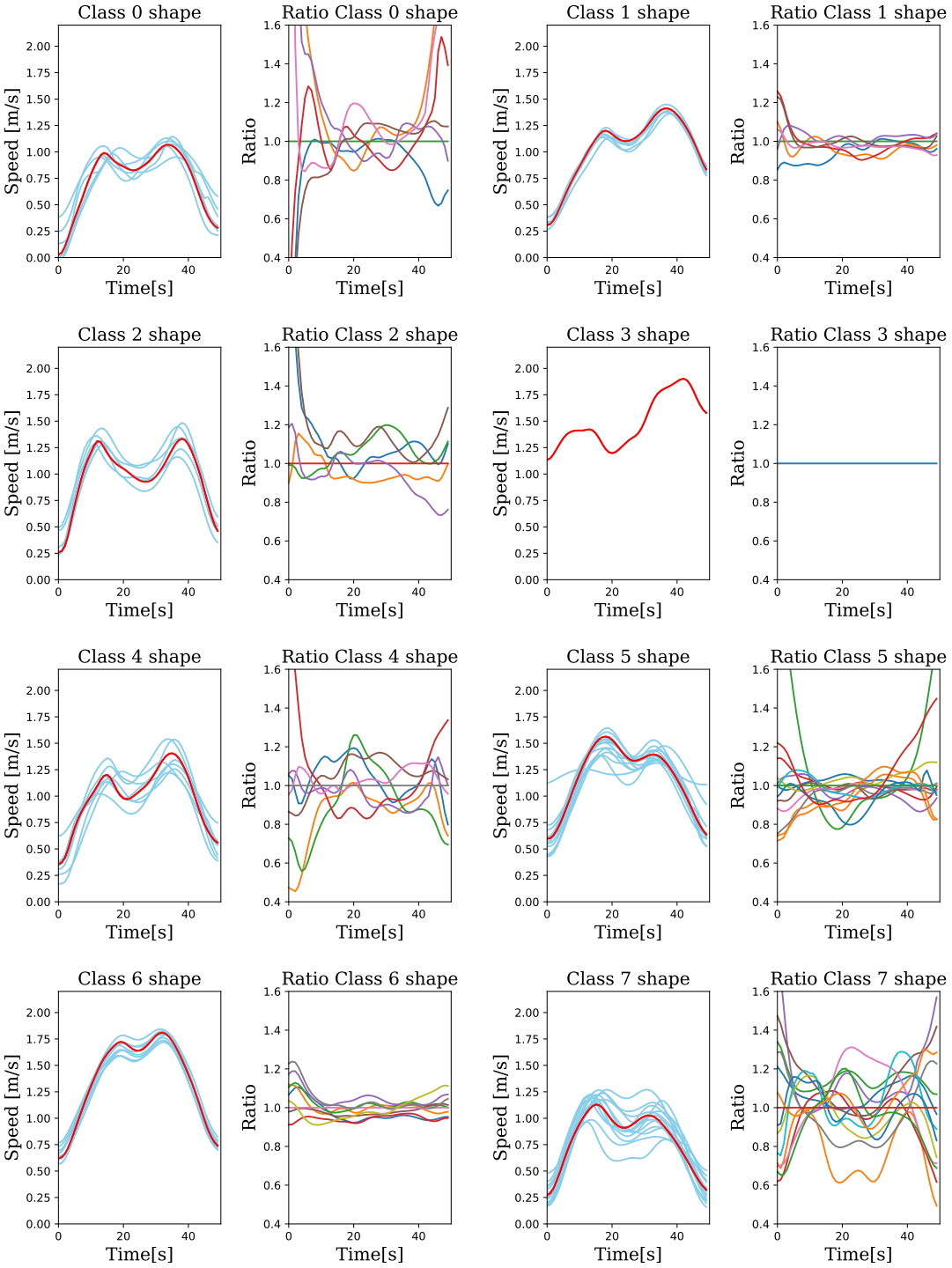

Figure 3.11: Mean absolute error. Plots of class shape (left) and ratio class shape (right).

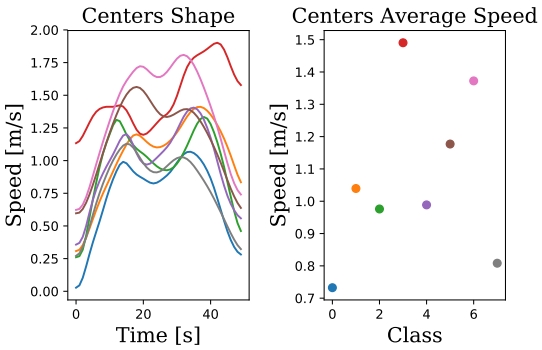

Figure 3.12: Mean absolute error. Plot of centers shape (left) and centers average speed (right)

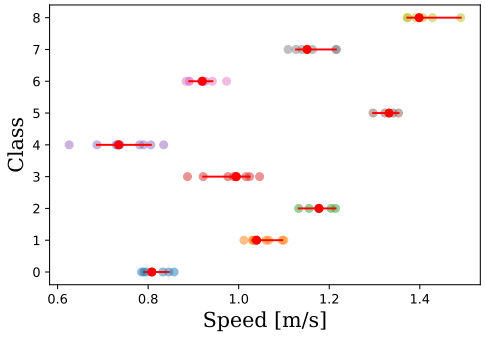

Figure 3.13: Kolmogorov-Smirnoff test. Median speed and std of each cluster (red).Average speed distribution of the elements of each clusters (transparent colors).

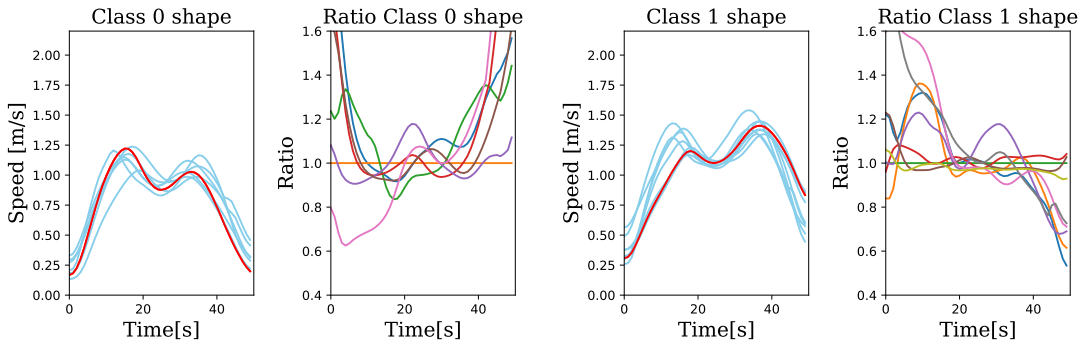

Figure 3.14: Kolmogorov-Smirnoff test. Plots of class shape (left) and ratio class shape (right) for class-0, class1.

in 3.14, 3.15. Class 0 contains seven elements with non similar shape. Class 1 contains nine elements with similar shape. Class 2 contains six elements with similar shape. Class 3 contains nine elements with non similar shape. Class 4 contains ten elements with non similar shape. Class 5 contains four elements with similar shape. Class 6 contains eight elements with similar shape. Class 7 contains seven elements with non similar shape. Class 8 contains five elements with similar shape and one elements with an anomalous shape. Indeed, as discussed later, the algorithm built with the Kolmogorov-Smirnoff test, has produced classes considering the value of their average speed .

3.6 Clusters analysis with zeros method

Another method to reconstruct swimming profiles with dimension is to add some zeros into the beginning of the speed vector (this method is called zeros method). Applying the affinity propagation algorithm to the swimming profiles constructed with the zeros method, different results in the clusters have been found. In particular, the number of classes obtained by the mean absolute error and the Kolmogorov-Smirnoff test is different from the one computed with the spline method. Indeed, 11 clusters have been produced with the mean absolute error rather

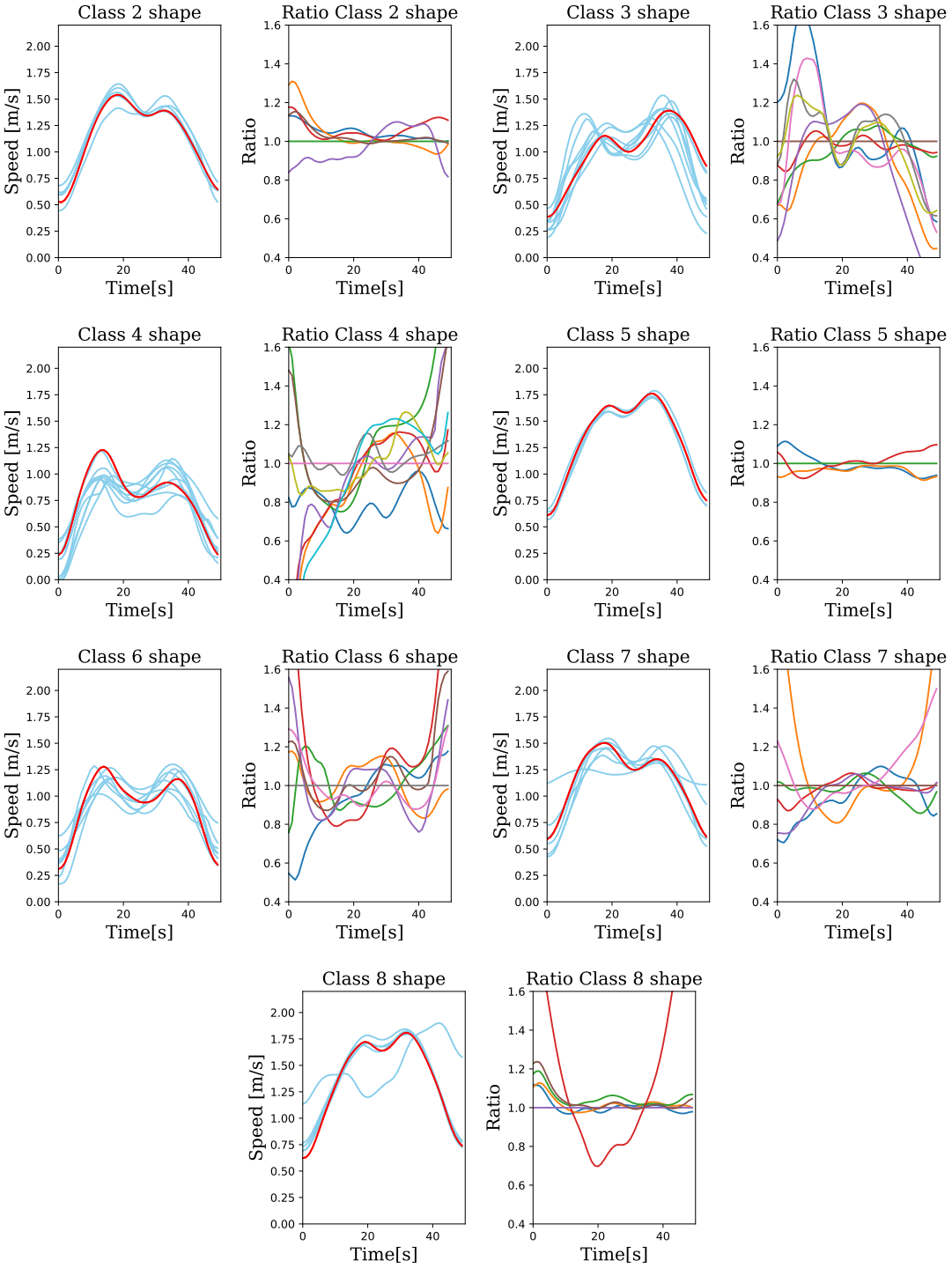

Figure 3.15: Kolmogorov-Smirnoff test. Plots of class shape (left) and ratio class shape (right) for class-2 to class-8.

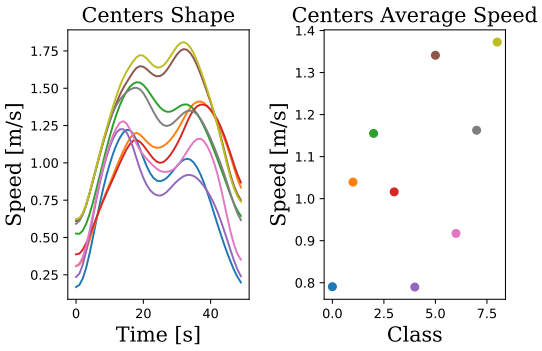

Figure 3.16: Kolmogorov-Smirnoff test. Plot of centers shape (left) and centers average speed (right).

then eight, and eight clusters have been produced with the Kolmogorov-Smirnoff test rather then nine. Instead, the euclidean distance and the mean squared error have produced the same number of clusters as the spline method.

In addiction, it has been impossible to identify the classes shape with the zeros method because each swimmer has the best swim in a different temporal position in the entire profile, but the zeros method requires to plot the original time. So it has been decided not to include this results in the present work.

3.7 Comparison of clusters results with spline method

In this section, the different results of the swimmers' profiles clustering are discussed. Three clusters have been produced, grouping the same elements by the euclidean distance, the mean squared error and the mean absolute error. The first of these clusters is shown class five with euclidean distance, class 3 with mean squared error, class 3 with mean absolute error and it represents the swimmer-1 who is the fastest one. The second of these clusters is class one with euclidean distance, class one with mean squared error, class six with mean absolute error and it

represents different performances of the same athlete named swimmer-2. The third of these clusters is class three with euclidean distance, class six with with mean squared error , class 1 with mean absolute error and it represents different performances of the swimmer-6. Based on these results, it is possible to underline that the affinity propagation algorithm recognizes the particular style of a single swimmer in different performances.

However, the main part of the dataset comprises single profiles of different swimmers (up to 35). Due to the high number of single swimmers' profiles, the algorithm does not recognize a particular pattern and creates different classes for each metric used. It is also possible to recognize some recursions in certain classes obtained by the different metrics. Unfortunately, these classes are composed by the profiles of different swimmers, hampering to identify a particular style. It is possible to speculate that these classes cluster elements with similar average speed and similar profile shape, as shown by the graphs plotting those features.

3.8 Analysis of swimmers' performance

A program which computes the value of all the metrics and the percentage loss between two swimmers has been built to compare the different performances. By this program, it is possible to analyse the performances of two different swimmers or to test the same swimmer in two different performances. To accomplish this analysis, the spline method has been used to reconstruct swimmers' best period profiles. In the subsections below, two examples of this analysis are reported.

3.8.1 Analysis of best and worst swimmers

The aim of this subsection is to study how the affinity propagation algorithm produces clusters of the fastest swimmers and the slowest

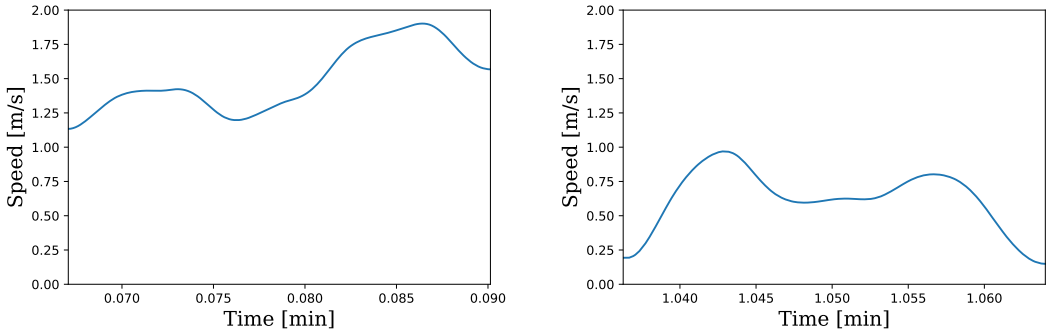

Figure 3.17: Best period of fastest (left) and slowest (right) swimmer

swimmers. The fastest swimmer is named swimmer-1 (3.17, left). He is clustered alone in graphs obtained by using the Euclidean distance, mean squared error, and mean absolute error. Instead, the Kolmogorov-Smirnoff test clusters him with other profiles (3.15, class 8), specifically with some of the repetitions of another swimmer named swimmer-2. Worthy of note is that those profiles of swimmer-2 are the fastest ones after swimmer-1' s ones.

The slowest swimmer is named swimmer-3 (3.17, right). The Euclidean distance and the squared mean error clusters him in the same class with other profiles which are the slowest ones after swimmer-3 (green dots in 3.4, 3.7). The mean absolute error clusters swimmer-3 in a different cluster with other swimmers which are not as slow as him (grey dots in. 3.10). The Kolmogorov-Smirnoff test clusters swimmer-3 in a class that comprises the slowest swimmers (violet dots in 3.13).

Analysing the shape of the classes of the fastest and the slowest swimmers, it is possible to detect the features which characterize the style of the fast swimmer versus the slow one. Indeed, in swimmer-3's profile, it is possible to observe a section between the two peaks where the swimmer does not gain speed. Instead, in swimmer-1's profile there is an important gain of speed after the first peak. Analysing the shape of the classes of the two swimmers, it is possible to observe the charateristic features quoted above.

3.8.2 Analysis of swimmer's performance with extra-weight

The aim of this subsection is to study swimmers' performances where the athletes carry extra weights. The dataset consists of three profiles of two different swimmers, named swimmer-4 and swimmer-5. In the three performances, the athletes carry weights (computed in Newton) respectively of zero Newton, 30 Newtons, and 50 Newtons. The results are shown in the graphs below. The speed percentage loss with increasing weight has been computed as the percentage ratio between the average speeds of two profiles.

As 3.18 shows, swimmer-4 loses the 7% of his speed carrying an extra weight of 30 Newtons and 21% increasing his weight from 30 Newtons to 50 Newtons. Instead, as shown in 3.19, swimmer-5 has lost only 10% of his speed increasing his weight of 30 Newtons, whereas 9% increasing his weight from 30 Newtons to 50 Newtons. Based on these results, swimmer-5 seems to be an athlete trained to swim with extra-weight. Indeed, he loses speed in a linear way when increasing the extra weight. Instead, swimmer-4 loses speed in an irregular way, when increasing the extra weight from 30 to 50 Newtons. This suggests that swimmer-4 could be an athlete who is not trained to swim with extra-weights. It is also possible to analyse how the affinity propagation algorithm clusters the profiles with an extra weight to check the loss in the swimming style instead of the speed loss.

The mean absolute error and the euclidean distance group swimmer-5's profiles of zero and 30 Newtons in the same cluster and swimmer-5's 50 Newton profile in a separate one. Instead, the mean squared error clusters swimmer-5's profiles of 30 and 50 Newtons in the same class, but it places swimmer-5's zero Newtons profile in another cluster. According to these results, it is possible to affirm that the affinity propagation algorithm does not recognize an important loss in the swimming style. All the metrics clusters swimmer-4's zero and 30 Newton profiles in the same class, whereas swimmer-4 with 50 Newtons in a separate cluster.

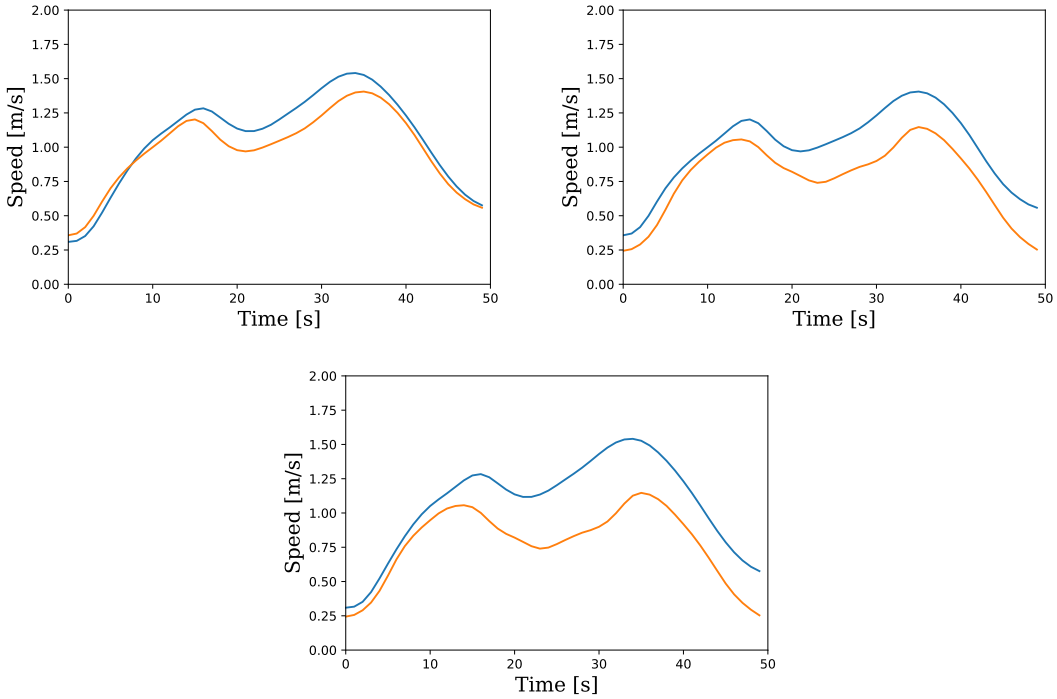

Figure 3.18: Swimmer-4 best period with extra-weight. Top left: 0N blue vs 30N yellow. Top right: 30N blue vs 50N yellow. Bottom left: 0N blue vs 50N yellow.

These results show that the algorithm has detected a significant loss in the swimming style as the percentage loss of 20% and 26% respectively suggests.

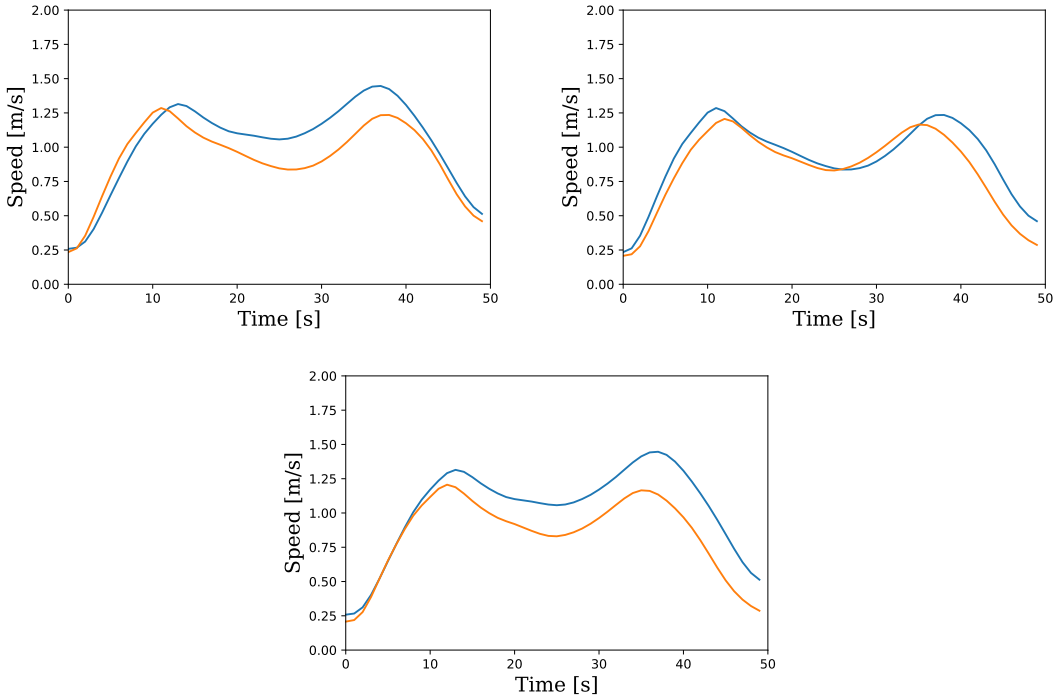

Figure 3.19: Swimmer-5 best period with extra-weight. Top left: 0N blue vs 30N yellow. Top right: 30N blue vs 50N yellow. Bottom left: 0N blue vs 50N yellow.

3.9 Anomalous profiles

Two anomalous profiles have been found in the analysis of the profiles. The first anomalous profile is shown in 3.20. It is considered anomalous because it does not record the succession of peaks representing the alternation of the stroke and the flutter kick. Two other profiles show this shape in the original dataset. It has been decided to remove those profiles from the dataset because the filtering program had difficulties to recognize that shape.

The second anomalous profile is the one that has been clustered alone using the euclidean distance, the mean squared error and the mean absolute error metrics. In addiction, the swimmer (swimmer-1) associated to that profile is the fastest one. It is possible to underline that it is an anomalous profile due to its unique shape, as it is possible to observe

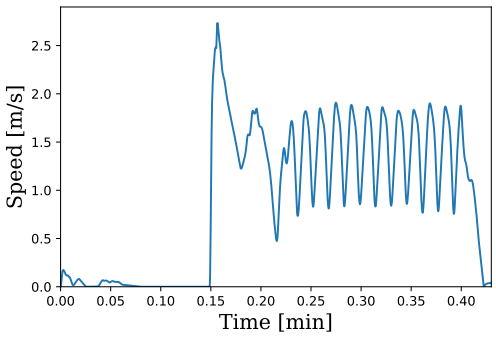

Figure 3.20: Anomalous profile

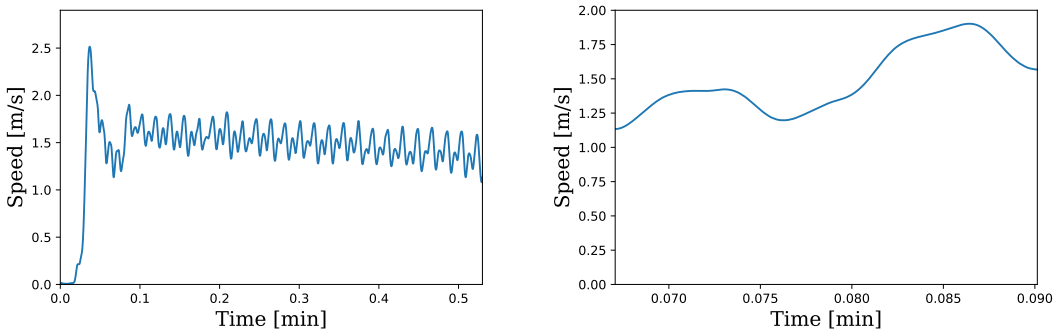

Figure 3.21: Swimmer-1 entire profiles (left) and best period (right)

in 3.21 (right). The question is if the filtering program has made some errors in filtering the entire profile. However, analysing in 3.21 (left), it is possible to observe that this particular shape represents the first the stroke and the flutter kick after the irregularities. So it is possible to conclude that the filtering program does not make any error in filtering.

3.10 Swimmer style discussion

Based on the performed analyses, it is possible to infer some conclusions about the swimming style. It has been verified that the flutter kick is the most powerful movement, as it has been described in the introduction. In fact, analysing the fastest swimmer's profile it is possible to observe

that tit is the one with the highest peak in correspondence of the flutter kick. Analysing 3.21 (left), it is possible to speculate that the fastest swimmer spends a lot of energy to swim, so its endurance could be very short. Indeed, the fastest swimmer has only a little speed drop after the flutter kick, making its performance very fast, but its endurance short. In other profiles, as those in 3.1, there is a big so drop after the flutter kick, so the average speed is lower, but they can endure the performance longer.

Chapter 4

Conclusions

In the present work, we have verified that the affinity propagation algorithm is able to recognise different performances of the same swimmer, clustering them in the same class. Furthermore, this result has been obtained using different metrics. It would now be interesting to apply the affinity propagation algorithm to a dataset composed by 8-10 swimmers with 5-6 different profiles each, to test if that algorithm is able to recognise the different athletes performances also in that case.

A method has been proposed to compare different performances of same swimmer or of several swimmers, confirming the importance of the flutter kick in attaining high breaststroke speed.

Finally, based on the results of this work, three kinds of specific training can be proposed.

1. The first is a training which aims to improve the swimming style trying to reproduce the profile of a better performing swimmer, in particular the distribution of speed in time. Comparing his profile with that of a better performing swimmer, an athlete could change its style and attempt new movements which may enhance his speed.

2. The second is a training where different swimmers try to reproduce the style of other swimmers, who may be also less performing than them. That training could allow the athletes to learn how to gain a

better control of their movements improving their style. In addition, a swimmer could understand the pros and cons of its style.

3. The last training proposal is to compare different styles in endurance tests on long term performances detecting the best ones. Understanding the features of the stronger swimmer, it is possible to create a particular training to strengthen the athletes. That strengthening could be also useful in short distance performances to improve the swimmers' speed.

Bibliography

[1] B. A. Novatchkov H., "Artificial intelligence in sports on the example of weight training," *Journal of Science and Medicine in Sport, 12(1)*, pp. 27–37, 2013.

[2] U. O. Mezyk E., "Machine learning approach to model sport training," *Computers in Human Behavior 27*, p. 1499–1506, 2011.

[3] C. M. J. B. T. M. Bartolomeu, R. F., "Contribution of limbs' actions to the four competitive swimming strokes: a nonlinear approach," *Journal of Sports Sciences, 36(16)*, pp. 1836–1845, 2018.

[4] C. D. A. M. M. J. S. A. J. Barbosa, T. M., *Biomechanics of Competitive Swimming Strokes. In V. Klika (Ed.), Biomechanics in Applications*. IntechOpen, 2011.

[5] E. W. Maglischo, *Swimming fastest*. Human Kinetics, 2003.

[6] L. H. H. R. K. J. B. C. C. D. Seifert, L. M., "Inter-individual variability in the upper-lower limb breaststroke coordination," *Human movement science, 30(3)*, pp. 550–565, 2011.

[7] S. L. M. C. D. Leblanc, H., "Arm-leg coordination in recreational and competitive breaststroke swimmers," *Journal of Science and Medicine in Sport, 12(3)*, pp. 352–356, 2009.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[9] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[10] C. Sammut and G. I. Webb, eds., *Mean Squared Error*, pp. 653–653. Boston, MA: Springer US, 2010.

[11] C. Sammut and G. I. Webb, eds., *Mean Absolute Error*, pp. 652–652. Boston, MA: Springer US, 2010.

[12] S. K. e R.A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics, 22(1)*, pp. 79–86, 1951.

[13] W. J. Marsaglia G, Tsang WW, "Evaluating kolmogorov's distribution," *Journal of Statistical Software, 8 (18)*, pp. 1–4, 2003.

[14] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online; accessed <today>].

[15] B. J. F. et al, "Clustering by passing messages between data points," *Science*, 2007.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

I am very grateful to prof. Stefano Carrazza for having offered me the opportunity to investigate this very interesting topic which combines a new and promise tool such as machine learning to a sport discipline, a subject that has not been studied up to now.
I am indebted to the athletes of Dipartimento di Scienze Sportive, whose fatigue has provided me some work.
I warmly thank my family for having supported me and thought me to work hard.
I thank all my friends and Miriana because they make my efforts lighter.