FILTERING PROCEDURES FOR SENSOR DATA IN BASKETBALL

Rodolfo Metulini*

SUMMARY

Big Data Analytics help team sports' managers in their decisions by processing a number of different kind of data. With the advent of Information Technologies, collecting, processing and storing big amounts of sport data in different form became possible. A problem that often arises when using sport data regards the need for automatic data cleaning procedures. In this paper we develop a data cleaning procedure for basketball which is based on players' trajectories. Starting from a data matrix that tracks the movements of the players on the court at different moments in the game, we propose an algorithm to automatically drop inactive moments making use of available sensor data. The algorithm also divides the game into sorted actions and labels them as offensive or defensive. The algorithm's parameters are validated using proper robustness checks.

Keywords: *Sports Statistics, Big Data, Sport Analytics, Human Activity Recognition.*

1. INTRODUCTION

Professional team sports' managers, more and more in recent years, are becoming aware of the potential of Data Analytics in order to better manage their team. In team sports, continuous interactions among three agents – coaches, single players and the whole team – produce an high level of complexity. This complexity has been studied, among others, in the new domain of ecological dynamics (Travassos, Davids, Araùjo and Esteves, 2013; Araùjo and Davids, 2016). Nowadays, Information Technologies (IT) make large amounts of real-time information on teams and players available. Most of the results from the interaction among these three agents could be captured by two kind of data: (i) play-by-play data (also called *event-log*), which report a sequence of relevant events that occur during a match, related to either the team or the single player, such as shots or fouls; (ii) the positioning, the velocity and the acceleration of players or the ball, also called sensor data, which is captured through Global Positioning System (GPS) techniques. There is high potential in jointly using these two kind of data to cope with the intrinsic complexity in team sports and with the aim of producing advanced statistics for team managers. Several aspects are already taken into account in the scientific literature, Gudmundsson and Horton (2017) being a nice and quite complete review. For example, Carpita, Sandri, Simonetto and Zuccolotto (2013, 2015) used data mining techniques in order to identify the drivers that mostly affect the probability to win a football

_____

* Dipartimento di Economia e Management - Università di Brescia - Contrada Santa Chiara, 50 - 25121 BRESCIA (e-mail: rodolfo.metulini@unibs.it).

match. Social network analysis has also been used to capture the interactions between players (Wasserman and Faust, 1994); Passos, Davids, Araùjo, Paz, Minguens and Mendes (2011) used centrality measures with the aim of identifying central players in water polo. A necessary condition to produce statistics is to correctly understand data, by collecting, storing and processing them in a proper way. A review on this regard has been made by Stein, Janetzko, Seebacher, Jager, Nagel, Holsch and Grossniklaus (2017).

This paper is about basketball data processing. Basketball is a sport played by two teams of five players each on a rectangular court. The objective is to shoot a ball through a hoop 46 centimetres in diameter and mounted at a height of 3.05 meters to backboards at each end of the court. According to International Basketball Federation (FIBA) rules, the match lasts 40 minutes, divided into four periods of 10 minutes each. There is a 2-minute break after the first quarter and after the third quarter of the match. After the first half, there is an half-time break.

The manuscript focuses on the processing of players' movements data. In particular, the aim of this manuscript is to i) automatically drop all the inactive moments from a data matrix that tracks the movements of the players on the court at different moments in the game, ii) automatically split the game into sorted actions, labelling them as offensive or defensive. To do that we make use of available sensor data tracked during a game. This work is similar to that by Wu and Bornn (2017), as they provide a procedure to process ball's and players' trajectories. However, our procedure differs, since it works even if data on ball's movement is missing. We place this piece of research within the domain of Human Activity Recognition (HAR). HAR aims to recognize the actions of an agent from a series of observations on the agents' actions and the environmental conditions, Trabelsi, Mohammed, Chamroukhi, Oukhellou and Amirat (2013) being a representative article on such a topic. In this work, the agents are considered to be the players of a team as a whole moving inside the court, and the action to be recognized concerns whether the game is active or inactive. In this vein, Huang, Chen, Chiu, Yi, Lin, Yeh and Kuo (2012) apply data automation algorithm in sports, by using sensor data to categorize golf swing trajectories. Jiang, Ye, Gao and Huang (2004) propose a game segmentation algorithm that suits in different sports. Jordan, Melouk and Perry (2009) propose a model based on design of experiments and response surface methodology.

In this paper we propose and discuss a multiple-stage algorithm which aims to drop inactive moments of a basketball data matrix tracking players' movements. We apply this algorithm to different real case studies (CS) in order to calibrate the algorithm's parameters by means of a data-driven approach. We then provide some descriptive statistics related to the CS for a validation check of the algorithm and of the robustness of the parameters.

In Section 2 we present and explain the algorithm. In Section 3 we validate the algorithm using real data. Section 4 concludes the paper and suggests further analysis.

Here we describe the algorithm aimed to automatically reduce a basketball data matrix with data on players' positions at different times to just the moments when the game is active, and to consistently split the game into sorted and labelled (offensive or defensive) actions, using tracked sensor data only. The algorithm is suitable in cases where i) information on the movement of players on the court has been captured with the use of appropriate GPS devices, for example, the accelerometer, a device that measures proper acceleration (Tinder, 2006) and ii) nobody, during the game, is in charge to take note of relevant informations of the game, such as active moments, offensive or defensive moments, and so on. Just to be clear, the only information available is that from GPS devices.

As accelerometers track information of players' movement along the full game, data consists of a total of around 90-100 minutes, despite only 40 of these are actually related to moments of active play, and therefore relevant to the aim of the analysis of players' movement. For this reason, the objective of the algorithm is to reduce game data to the moments when the game is active (40 minutes). A parallel objective is to split the game into actions, in such a way the first (in chronological order) action is identified with a 1, the second one with a 2, and so on. In doing that, we obtain a reduced data matrix having a correct number of actions (each with a correct length duration).

The algorithm applies to a data matrix $\mathbf{X}$ where each row corresponds to a sensor record, described by the variables time (in milliseconds, ms). $\mathbf{X}$ should be sorted from the smallest ms to the largest ms. Each row contains information on several variables related to the positioning (pos) and the velocity (vel) of each single player of one team ($p_1, p_2, ..., p_k$), along the court length (x), and the court width (y). Table 1 reports a sample of $\mathbf{X}$. It is important to clarify that, with the accelerometer, data are detected with a non-constant frequency; in addition, data of different players are recorded at different time instants. The dataset should contain any detected time instant. To do that, is necessary to attribute the last datum available to players not detected in that time instant. This procedure will be described in detail in Section 3.

TABLE 1. - *Sample of a subset of the data matrix* $\mathbf{X}$

ms	pos_{p_1-x}	pos_{p_1-y}	vel_{p_1-x}	vel_{p_1-y}	pos_{p_2-x}	pos_{p_2-y}	vel_{p_2-x}	vel_{p_2-y}
5564	4.28	7.40	1.26	1.26	15.25	8.98	0	0
5579	0.32	1.03	0.36	0.36	15.25	8.98	0	0

The algorithm consists of two main parts. In the first part it removes rows from $\mathbf{X}$ according to three different criteria. All these criteria are based on players' positioning and velocity. According to the first criterion the algorithm drops, from $\mathbf{X}$, the instants in which the number of players inside the court is different from five. With this criterion we ideally remove all the moments related to pre-match and post-

match periods, half-time and quarter-time intervals, time-outs and so on. The second criterion aims to drop, from **X**, the instants in which a player is shooting a free throw, by looking to his positioning in the court. The third criterion aims to remove those moments where all the five players in the court report a velocity lower than h_2 km/h, for at least h_3 consecutive seconds, where h_2 and h_3 are subject to a parameter calibration. The second part of the algorithm assigns actions' sorting and labelling to the reduced data matrix, by looking to the average positioning of the five players on the court.

We now describe the steps of the algorithm in detail, as also schematically summarized in Figure 1.

- In the step 1-A th algorithm drops the rows where the number of players in the court is different from five. To do that, the algorithm creates a new variable for each of the k players. For player 1 (p_1), this variable assumes value 1 when the player's coordinates ($pos_{p_1\_x}$ and $pos_{p_1\_y}$) are inside the court, 0 otherwise. The algorithm, after having computed this variable for all the k players, creates a new variable (*count*), which is the row sum of the k 0/1 variables. If *count* $= 5$, it means that, in that specific *ms*, the number of players in the court is exactly five. The algorithm removes the rows where *count* $\neq 5$ (See Figure 2).

- In the step 1-B the algorithm drops the rows that correspond to the moments in which a player is shooting a free throw. To do that the algorithm generates a new variable (*ft*) which assumes value 1 when at least one player's coordinates lie inside the free throw circle. The algorithm assumes that a player is shooting a free throw when he remains inside the circle for at least $h_1 = 10$ consecutive seconds[1]. When *ft* consecutively reports value 1 for at least 10 seconds, the algorithm drops all the corresponding rows from the data matrix (Figure 3).

- In the step 1-C the algorithm drops the rows corresponding to moments where all the five players in the court are not running, for a certain number of consecutive seconds. This further step is necessary because steps 1-A and 1-B do not completely filter out inactive moments. For example, the moments in which the referee whistles for a foul, the moments when the ball comes back into play, or the moments of players' change are not detected in the first two steps. Moreover, since trajectories are available for one team only, step 1-B drops moments in which a player from that team is shooting a free throw, while step 1-C should also drops moments when a free throw has been attempted by a player of the opponent team. To do that, the algorithm first computes the velocity of each player, being $vel_{p_1} = \sqrt{vel_{p_1\_x}^2 + vel_{p_1\_y}^2}$ the velocity of player 1. There are moments where all the five players' velocity is less than h_2 km/h for at least h_3 seconds. Those are the moments to be dropped. Thinking to a real game, we assume a feasible range for the parameter h_2 to be larger than 8, whereas this measure is expressed in km/h; in fact, a walking player does not exceed 8 km/h. Passing to

---

[1] We set parameter h_1 as a constant. However, a tuning could be apply on this parameter.

h_3, a feasible range for this parameter is larger than 1, as there could be active moments (lower than 1 second) where nobody is running.
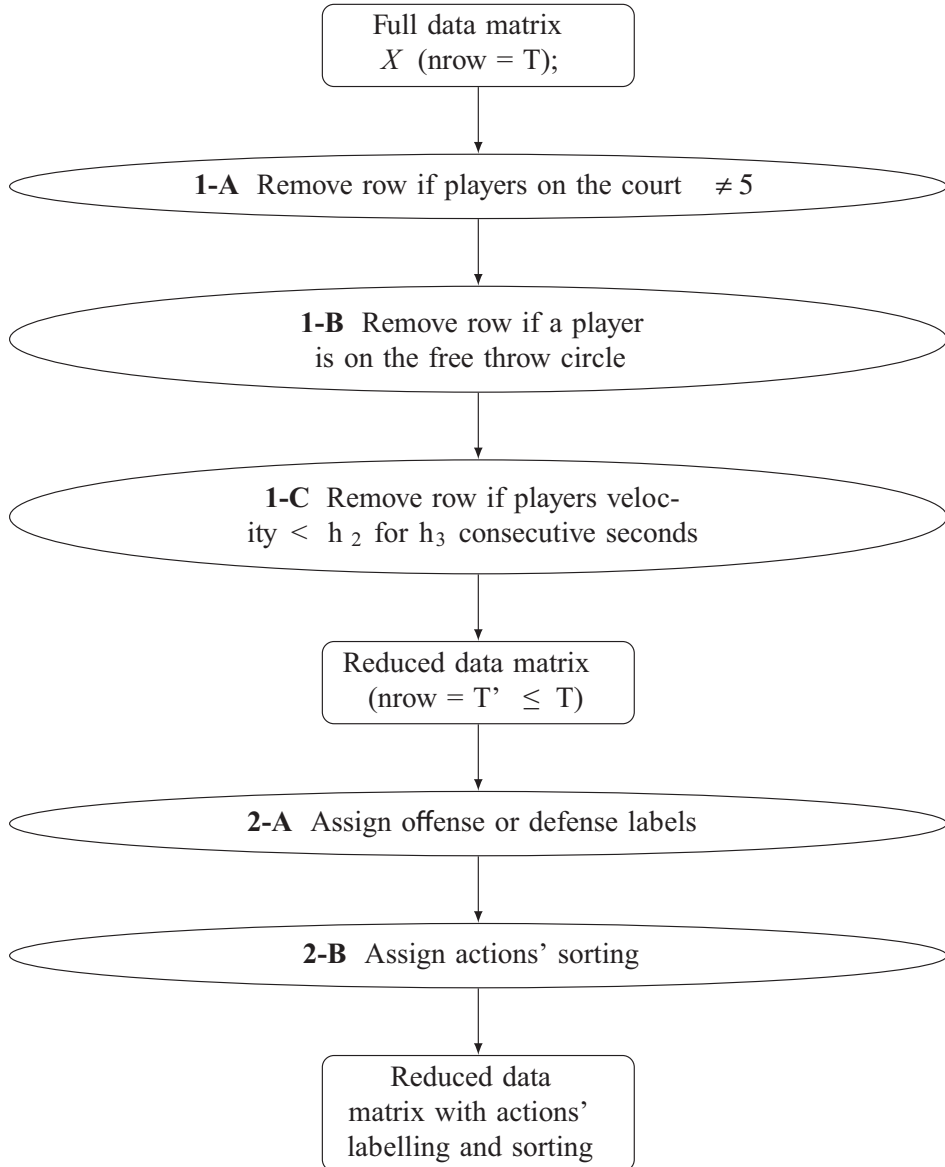

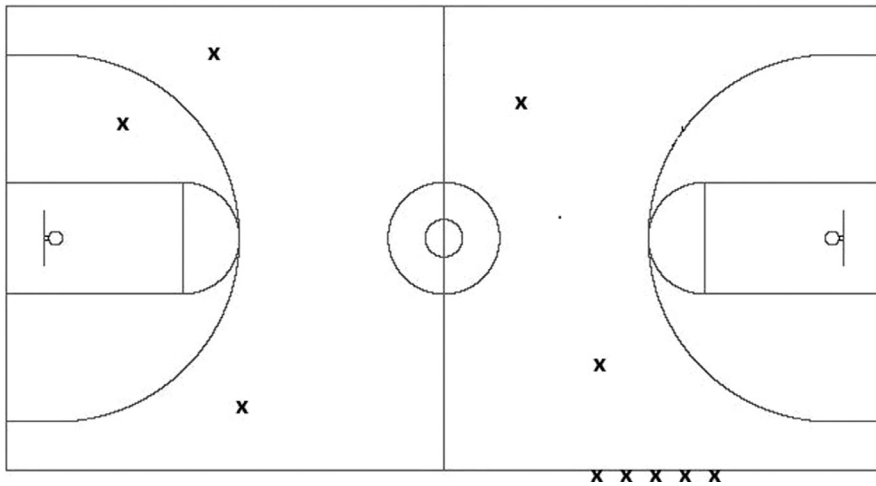FIGURE 1. - *Flow chart representing the steps of the algorithm*

FIGURE 2. - *Step 1-A of the algorithm. The figure represents a moment in which exactly five players are inside the court*
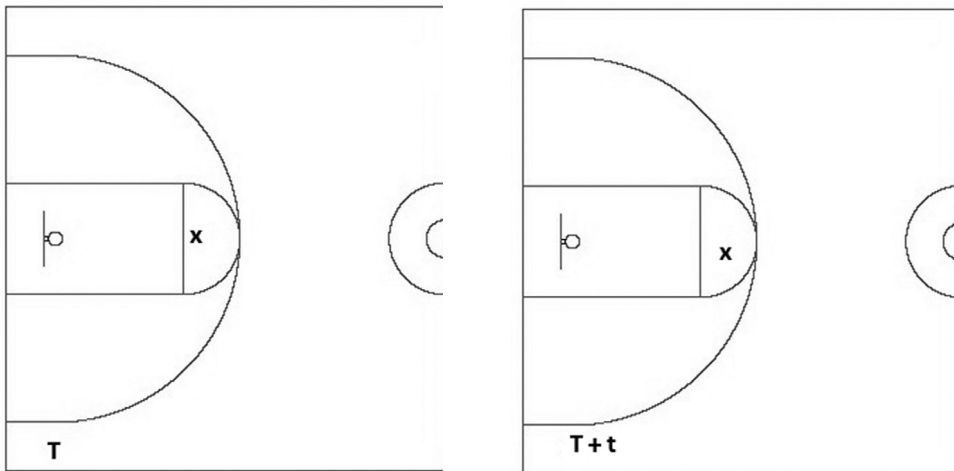

FIGURE 3. - *Step 1-B of the algorithm: the figure represents the criterion used to drop moments in which a player lies inside the free trhrow circle*

After these three steps, the algorithm generates a reduced version of the data matrix, which should include information for about 40 minutes of game. The reduced version is then processed throughout these two additional steps:

● The step 2-A aims at assigning a label to each row of the data matrix. The label regards whether the row belongs to a moment where the team is in offense or in defense. In doing that, the algorithm generates a new variable on the reduced

version of the matrix $\mathbf{X}$, *avg_pos*, that represents the average coordinate of the five players on the court, where *avg_pos* is a vector of two elements $[avg\_pos_x; avg\_pos_y]$, being $avg\_pos_x = \frac{\sum_{i=1}^{5} pos_{p_i-x}}{5}$ and $avg\_pos_y = \frac{\sum_{i=1}^{5} pos_{p_i-y}}{5}$. *avg_pos* could lies either on the offensive or on the defensive side of the court (see Figure 4). The algorithm also labels transition, that corresponds to those moments having avg_pos_x in the interval [+4,-4] meters (*m*) from the half-court line. The step ends by generating a new variable that assumes either value *O* (offense), *D* (defense) or *Tr* (transition)[2].
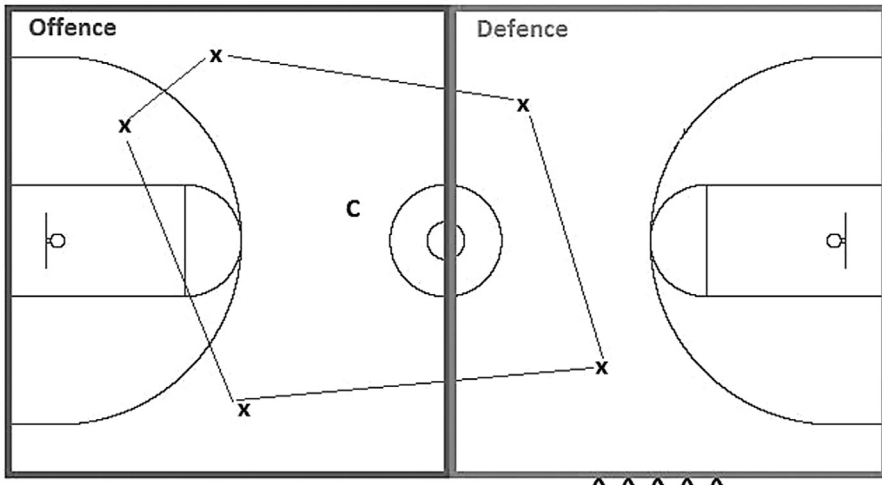

FIGURE 4. - *Step 2-A of the algorithm: the figure represents the criterion used to assign "offense" or "defense" labels*

- In the step 2-B the algorithm attributes to each row its action number, so that the moments related to the first action report value 1, the moments of the second action report value 2, and so on. We adopt the following criterion: the algorithm creates a new variable (*act_id*) that reports value 1 for the row with the smallest value of *ms*. As the data matrix is sorted from the smallest *ms* to the largest *ms*, a subsequent row *t* belongs to action 2 (i.e increases by 1) if avg_pos_x of the previous row $t-1$ is on one side of the court and avg_pos_x in *t* is on the other side. In doing that, we adopt a correction: *act_id* increases by 1 only if avg_pos_x, from *t* to $t+1$, passes over the transition area represented by the interval [+4,-4] meters from the half-court line (coloured area in Figure 5).

---

[2] The algorithm takes into consideration that after the half-time break the two teams change court side.
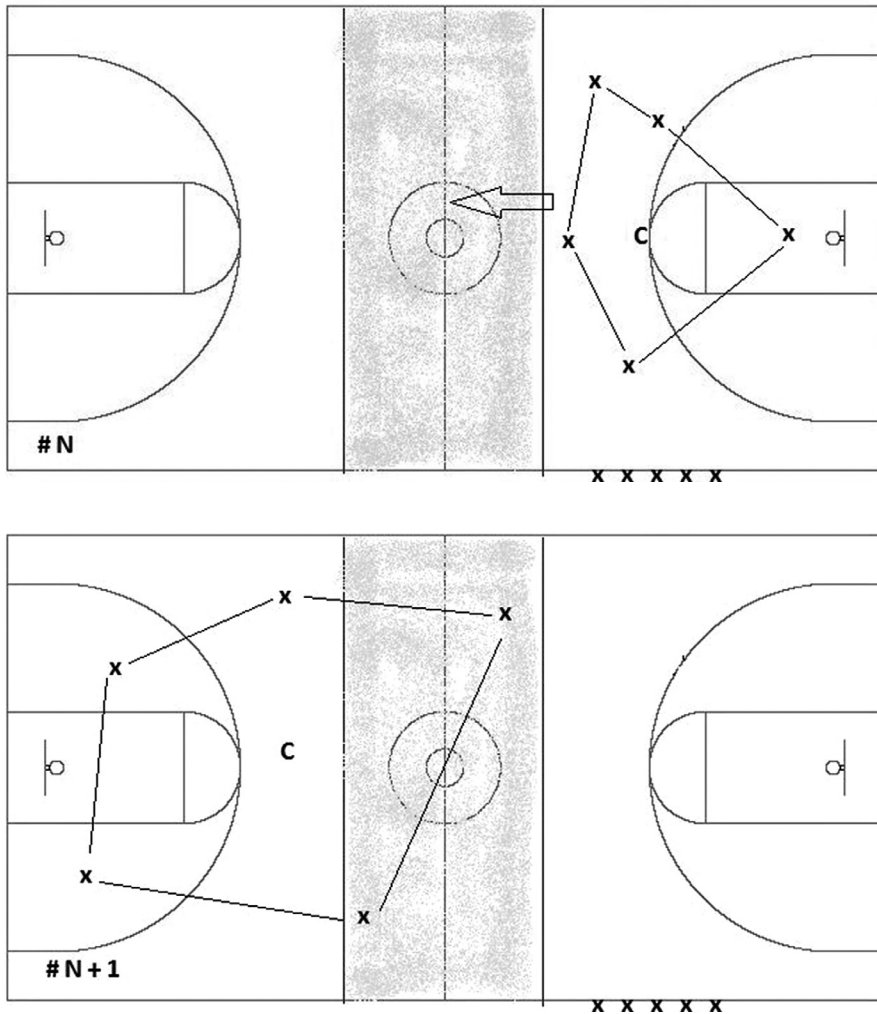
FIGURE 5. - *Step 2-B of the algorithm: figures represent the criterion used to assign moments to consecutive actions*

After these two additional steps, the algorithm produces a reduced version of the data matrix **X** with action labelling and sorting. This generates a cleaned and ready-to-use data matrix that can be used for producing various kind of advanced statistics.

3. EMPIRICAL APPLICATION

In this section we test the algorithm on real basketball games. We also perform a calibration analysis for the choice of the parameters used in the step 1-C of the algorithm. After having applied it to the CS, we present some descriptive analysis to evaluate the

performance of the algorithm in relation to specific elements that characterize the game, such as the number of actions and their duration. First of all, we present data and describe preliminary data processing that is needed before running the algorithm.

3.1 *Data*

Data refer to three games played by Italian professional basketball teams, at the Italian Basketball Cup Final Eight. MYagonism (https://www.myagonism.com/) was in charge to set up a system to capture these data during the games, trough accelerometer devices. Each player worn a microchip that, having been connected with machines built around the court, collected the player's position (in pixels of 1 cm^2 size) in the x-axis (court length), the y-axis (court width), and in the z-axis (how high the player jumps). Data have been detected with an average frequency of about 80 Hz. The initial data matrices contain information on players' positioning, velocity and acceleration during the full game length. Throughout the text, we will call the three games, Case Study 1 (CS1), Case Study 2 (CS2) and Case Study 3 (CS3)

3.2 *Data Processing*

Data have been provided to us in the form of several .csv files, each containing data of a single player in a single game. Each file is named with a univocal player's code, and a code-name concordance table has been provided. Each file contains three variables: *label*, *ms* and *value*. label refers to the name of the captured information, that can be positioning, velocity or acceleration, in the x-, y- or z-axis. ms reports the millisecond while value reports the information. Before the algorithm, preliminary data processing is needed.

- After having manually converted .csv files in .xls format, and having renamed files with the player's name, we generate several .txt files, each for a single player in a single game. These new files contains five columns: *label*, *ms*, *value*, *names*, *team*. *names* reports the player's name. *team* reports the team's name and the date of the game;
- we manipulate the files generated at the previous step in such a way the output file reports the variables positioning, acceleration and velocity, in x-, y- and z-axis, for each millisecond (variable *ms*) in which at least the data of one player has been captured. The data refer, in case a player has not been tracked in that millisecond, to the most recent available *ms*;
- z-axis and acceleration variables has been dropped, because not useful to the running of the algorithm. Then, *time* (an identifier for sorted time instants) and *id* (an identifier for the player) variables have been created. At this stage of the preliminary data processing, the data matrix is structured in order to draw motion charts, as described in Metulini (2017);
- in the end, the data matrix has to be reshaped in such a way each player's vari-

ables are reported in column. This data matrix is in the structure to be used as input for the algorithm. Positioning is expressed in *meters.centimeters* from the half-court line, while velocity is expressed in meters per seconds (m/s).

A number of additional variables has been computed. These new variables are instrumental to validate the algorithm.

In detail, we generate:

- the list $d_1, ... , d_{n^2-n}$ of $n^2 - n$ variables reporting the distance (in m) between players' pairs, being the distance between player i and player j $d_{ij} = \sqrt{(pos_{p_i\_x} - pos_{p_j\_x})^2 + (pos_{p_i\_y} - pos_{p_j\_y})^2}$; the value of d_{ij} is missing if at least one of the two players is on the bench in that specific *ms*;

- the variable d_{avg}, which reports the average distance (in m) among all the $n^2 - n$ distances ($d_{avg} = \frac{\sum_{i=1}^{n^2-n} d_i}{n^2-n}$);

- the variable *con_hull*, that reports the area (in m^2) occupied by the five players on the court, also called convex hull area;

- the variable vel_{avg}, which is the average velocity of the five players in the court ($vel_{avg} = \frac{\sum_{i=1}^{5} vel_i}{5}$).

We replicate this data manipulation procedure for each of the three CS.

3.3 Parameters' Validation

Recalling that the aim of the algorithm is to reduce the data matrix to a total of 40 minutes of game, we explore different combinations for the parameters h_2 and h_3 in relation to the resulting number of minutes. Minutes have been computed by comparing *ms* in consecutive rows ($ms_t - ms_{t-1}$) in the non-reduced data matrix that enters the algorithm. As *ms* is expressed in milliseconds, a proper conversion to minutes has made.

Moreover, since different games involve different team-mates and opponents, it is likely that the optimal combination for the parameters h_2 and h_3 slightly changes over CS1, CS2 and CS3. However, we do not want to find precise values, but a range of acceptable values.

Figure 6 displays contour plots for the three CS. The chart reports the contours levels for the length (in minutes) of the reduced data matrix, according to different combination of h_2 and h_3 parameters. The contours are evaluated on the range [8;11] according to the parameter h_2 and on the range [1;4] for h_3 parameter. We note that, for all the three CS, game length increases with the increase of h_3 parameter, while it decreases to an increase on the h_2 parameter. Interestingly, we find similar evidence over different CS, in terms of the choice of the parameters. Consistently over the three CS, the algorithm performs better when h_2 lies in the interval [9;10] and h_3 lies in the interval [2;3].
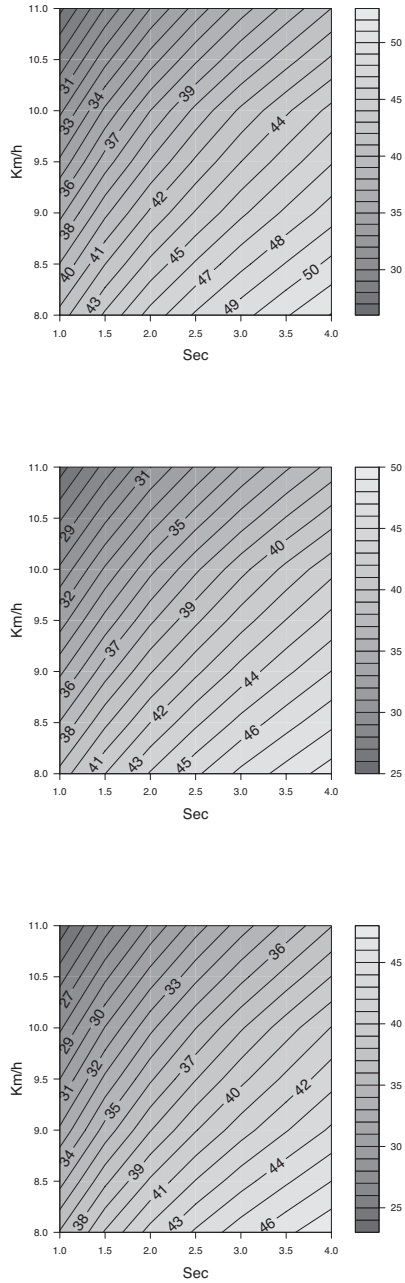
FIGURE 6. - *Contour plot representing the effective minutes of game obtained by the algorithm, subject to different combination of parameters h_2 (km/h) and h_3 (seconds) combination. From top to bottom: CS1, CS2 and CS3*

3.4 Results

In this section we present the results of the algorithm for the three real case studies in terms of relevant game characteristics, such as the game length, the number of actions and the action's length distribution, as well as the average distances among team-mates and the area occupied by them. These characteristics could depend on coaches' tactics, players' strength and team-mates cohesion, but they may also depend on whether the action is an offensive or a defensive one, on the presence of a specific player or on a specific line-up in the court. Comparing game characteristics with those generally obtained in real games, serves as a test for the robustness of the algorithm. Based on the indication given by the parameter validation in the previous subsection, we choose $h_2 = 9$ km/h and $h_3 = 2.5$ seconds for CS1, $h_2 = 9.4$ km/h and $h_3 = 2.5$ seconds for CS2, $h_2 = 10$ km/h and $h_3 = 2.5$ seconds for CS3.

We start by characterizing the three games in terms of players' distances and surface area occupied by team-mates (convex hulls area). We report analyses separately for offensive and defensive instants, as Metulini, Manisera and Zuccolotto (2017) found that surface area significantly differs from offensive to defensive actions. Consistently with the results in the aforementioned work, average distance among players (d_{avg}), along the three case studies, is generally larger during offensive instants. Left panel of Figure 7 reports as illustrative example the related distribution for CS1. In defense, the average distance among payers reports a mean of 6.17 m and a median of 5.57 m. In offense, the average distance reports a mean of 7.96 m and a median of 7.97 m. With reference to surface area, results give a similar information. Convex hulls area, along the three case studies, are generally larger during offense. Right panel of Figure 7 reports as illustrative example the related distribution for CS1. In defense, the convex hulls area reports a mean of 32.29 m^2 and a median of 24.97 m^2. In offense, the convex hulls area reports a mean of 53.46 m^2 and a median of 51.98 m^2.
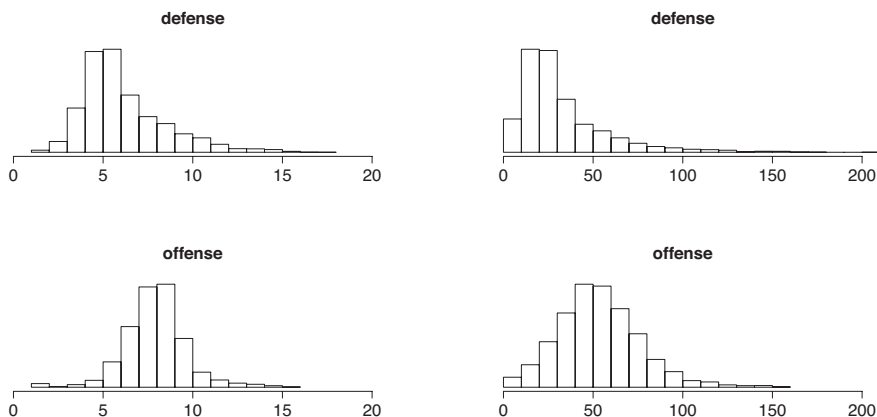

FIGURE 7. - *Histogram distribution of average distances (in m, left) and average convex hull areas (in m², right) during offense and during defense, in CS1*

Alike, we computed the average velocity (vel_{avg}) of the five players on the court. Figure 8 reports the related distribution for CS1. In defense, the players' velocity reports a mean of 5.56 km/h and a median of 5.10 km/h. In offense, the players' velocity reports a mean of 5.76 km/h and a median of 5.44 km/h. Moreover, players' velocity is larger than 6.71 km/h for the 25% of the defensive moments, larger than 7.07 km/h for the 25% of the offensive moments.
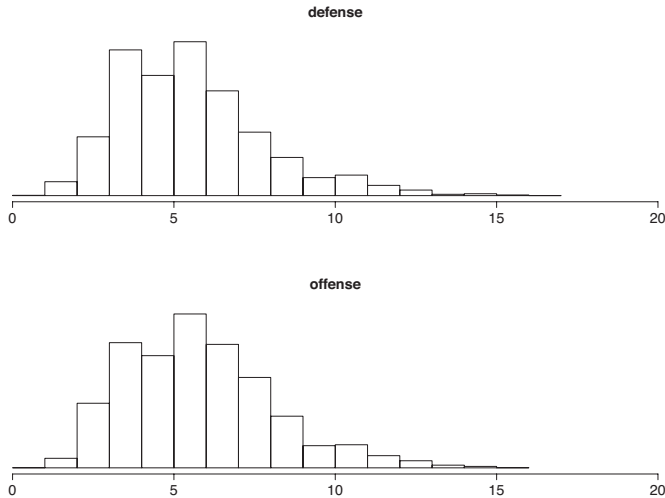

FIGURE 8. - *Histogram distribution of average velocity (in km/h) during offense and during defense, in CS1*

We now introduce the distributions of the number of actions retrieved by the procedure. The algorithm, applied to real case studies, divides the game into a consistent number of actions. The algorithm splits the games in a number of 151 actions for CS1, 136 actions for CS2 and 132 for CS3.

A robustness check consists on counting the number of actions that last for a reasonable time. Choosing an interval between 4 and 38 seconds[3], in CS1 147 out of 151 actions are included in this interval (97.4%). This number is 133, out of 136 (97.8%), for CS2 and 128, out of 132 (97.0%), for CS3. All in all, it emerges that most of the actions lie in a reasonable interval of duration, consistently along all the three games analysed.

The histogram in Figure 9 reports the distribution of the actions' duration in CS1. This histogram shows that actions last for a reasonable interval of time. The average duration stands to 16.00 seconds, the median stands to 15.66. Furthermore, the

---

[3] 38 seconds, due to the new rule that gives 14 additional seconds after a foul, and also considering a double ball possession after an offensive rebound.

23.18% of the actions last for less then 10 seconds, the 50.33% of them last for 10 to 20 seconds and the 26.49% last for more than 20 seconds.
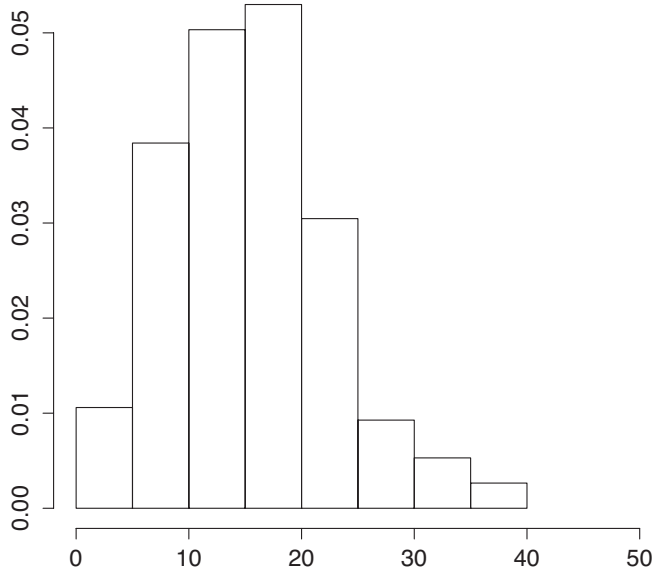

FIGURE 9. - *Histogram distribution representing the action duration (in seconds), in CS1*

Does the algorithm produce good results in terms of retrieving the correct number of actions in a game and the correct action duration? Until now, we have described the results obtained by applying the algorithm to three real case studies, and by evaluating them with respect to the rule of common sense.

Now, in order to further validate the procedure, we compare values from applying the algorithm to real CS with true information on real games. These information has been retrieved by looking to the web-scraped play-by-play of several games from both 2016 FIBA Olympics and 2015-2016 Italian professional A2 tournament. In relation to a bunch of matches, we report the distribution of the number of actions per game and the distribution of the actions' duration. More specifically, we use, as an example, one national team (USA) during the Olympic games and one Italian team (Leonessa Brescia) from A2. Histograms reported in Figure 10 refer to aggregated information from several games (i.e. the national team played 8 times during the Olympic games, A2 tournament lasts for 30 games.). Other teams of the same leagues displays similar results.
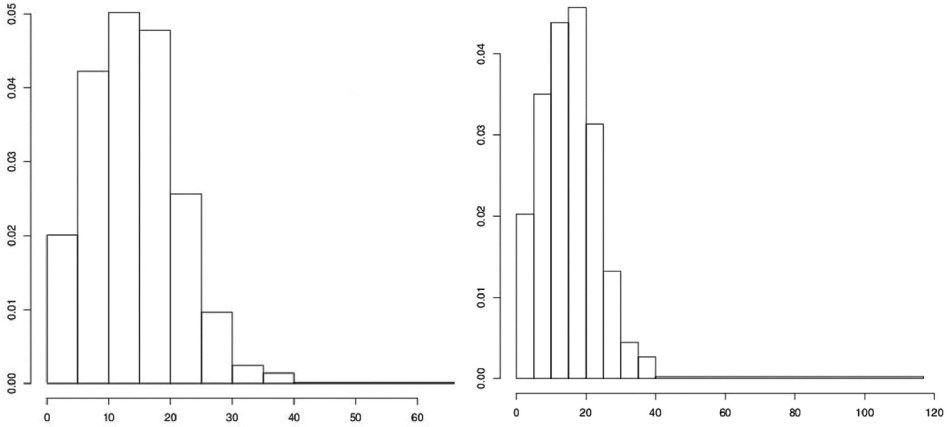
FIGURE 10. - *Action duration for team USA during Olympic Games (left) and for team Leonessa Brescia (Right). Data reports informations on aggregated games*

Left panel of Figure 10 displays the distribution of the actions' duration (in seconds) for team USA, while right panel reports the same distribution for team Leonessa Brescia. The two distributions are similar. The median duration for team Leonessa Brescia stands to 16. The median duration for team USA stands to 14. Moreover, the distributions present only few actions with a duration smaller than 10 seconds (31.60% for team USA, 27.50% for team Leonessa Brescia) or larger then 20 seconds (19.40% for team USA, 25.20% for team Leonessa Brescia), while most of the actions last for 10 to 20 seconds (49.00% for team USA, 47.30% for team Leonessa Brescia). These values are similar to those from applying the algorithm to CS1.

Left panel of Figure 11 displays the distribution of the number of actions in a single game, for team USA. Right panel reports the same distribution for team Leonessa Brescia. For the latter, the game lasted 140 to 150 actions 17 out of 29 times (58.62%). For team USA, the game lasted 150 to 170 actions 7 out of 8 times (87.5%).
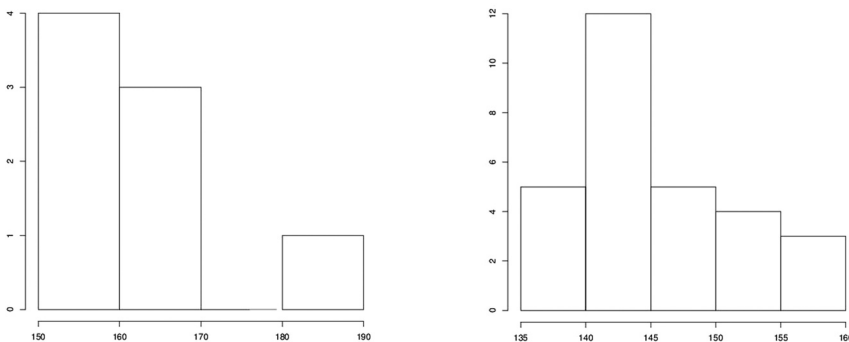

FIGURE 11. - *Number of actions for team USA during Olympic Games (Left) and for team Leonessa Brescia (Right). Data reports informations on aggregated games*

All in all, these values are consistent with those obtained by applying the algorithm to the real CS and demonstrate the robustness of our procedure.

4. DISCUSSION AND CONCLUSIONS

In the era of the Information Technology and Big Data Analytics, team sports' managers benefit from the availability of advanced statistics. However, statistics are just a tip in the iceberg, while there is an hard work behind, which concerns the steps of tracking, collecting, storing and processing the data. This paper concerned with basketball data processing, and aimed to suggest an ad-hoc procedure to automatically filter a data matrix containing players' movement information to the moments in which the game is active, and by dividing the game into sorted and labelled actions. In this regard, we placed this work within the area of the Human Activity Recognition, as we used players' actions to recognize a specific game state (i.e. inactive game moments). The algorithm has been tested on three different real games, and a series of robustness checks has been done, including a validation for the parameters to be used in the algorithm. Results of the validation suggests a stability of the two parameters along different games.

Practitioners which are in possession of a data matrix as the one described in this paper can replicate this procedure to analyse basketball matches.

The novelty of this procedure is that, unlike existing works, for example Wu and Bornn (2017), it works when the ball's trajectory is unavailable. However, further research is to be planned in order to validate the algorithm also with respect to a visual analysis of the same match.

REFERENCES

Araùjo D., Davids K. (2016). Team synergies in sport: theory and measures. *Frontiers in psychology*, **7**, 1449.

Carpita M., Sandri M., Simonetto A., Zuccolotto P. (2013). Football mining with R. In Z.Yanchang and C. Yonghva (Eds.), *Data Mining Applications with R*, (pp. 397-433). Elsevier.

Carpita M., Sandri M., Simonetto A., Zuccolotto P. (2015). Discovering the drivers of football match outcomes with data mining. *Quality Technology & Quantitative Management*, **12**(4), 561-577.

Gudmundsson J., Horton M. (2017). Spatio-temporal analysis of team sports. *ACM Computing Surveys (CSUR)*, **50**(2), 22:1 - 22:34.

Huang Y.C., Chen T.L., Chiu B.C., Yi C.W., Lin C.W., Yeh Y.J., Kuo L.C. (2012). Calculate golf swing trajectories from imu sensing data. In *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on* (pp. 505-513). IEEE.

Jiang S., Ye Q., Gao W., Huang T. (2004). A new method to segment playfield and its applications in match analysis in sports video. In *Proceedings of the 12th annual ACM international conference on Multimedia* (pp. 292-295). ACM.

Jordan J.D., Melouk S.H., Perry M.B. (2009). Optimizing football game play calling. *Journal of Quantitative Analysis in Sports*, **5**(2) 1-34.

Metulini R. (2017). Spatio-temporal movements in team sports: a visualization approach using motion charts. *Electronic Journal of Applied Statistical Analysis*, **10**(3) 809-831.

Metulini R., Manisera M., Zuccolotto P. (2017). Sensor Analytics in Basketball. *Proceedings of the 6th International Conference on Mathematics in Sports*. Padova University Press.

Passos P., Davids K., Araùjo D., Paz N., Minguns J., Mendes J. (2011). Networks as a novel tool for studying team ball sports as complex social systems. *Journal of Science and Medicine in Sport*, **14**(2), 170-176.

Stein M., Janetzko H., Seebacher D., Jager A., Nagel M., Holsch J., Grossniklaus M. (2017). How to make sense of team sport data: From acquisition to data modeling and research aspects. *Data*, **2**(1), 2 1-23.

Tinder R.F. (2006). Relativistic flight mechanics and space travel. *Synthesis lectures on engineering*, **1**(1), 1-140.

Trabelsi D., Mohammed S., Chamroukhi F., Oukhellou L., Amirat Y. (2013). An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *IEEE Transactions on automation science and engineering*, **10**(3), 829-835.

Travassos B., Davids K., Araùjo D., Esteves T.P. (2013). Performance analysis in team sports:

Advances from an Ecological Dynamics approach. *International Journal of Performance Analysis in Sport*, **13**(1), 83-95.

Wasserman S., Faust K. (1994). *Social network analysis: Methods and applications (Vol. 8)*. Cambridge university press.

Wu S., Bornn L. (2017). Modeling offensive player movement in professional basketball. *The American Statistician*, **72**(1), 72-79.