# **S1 Methods**

**Measurements in a controlled experiment.** Each trial began at the initial position fixed by the coach with a pass from the defender to the dribbler, and the trial finished with a successful shot, rebound by a defender or turnover (i.e., out of bounds or defensive team holding the ball). In the event of a defensive foul or out of bounds, the offensive team started attacking again with the ball (i.e., a reset trial which was excluded from analysis). The game was played with two referees according to the rules of basketball, with the exception of the specific rules stated above.

Four reflective markers were placed on each player’s body (top and left of the head and on their right and left shoulders). We used the midpoint of shoulders as the representative point of the players. We also used a basketball pasted a reflective sheet in a striped pattern to improve ball handling and recorded its position. All raw coordinate data points were smoothed using a fourth-order Butterworth low-pass digital filter (6 Hz).

**Scene selection.** We constructed an automatic play-detection system using the positional data. First, we categorized the state of the ball as holding, passing and shooting. (1) If the ball and an attacker was near (horizontal distance was less than 1 m) and horizontal ball acceleration was less than –5 m/s2 during a ball catch (i.e., the ball was in the air), the state of the ball was defined as the holding. The involved attacker was defined as the ball attacker. (2) If the ball was not holding, ball vertical position was adequately above the ring (more than 3.2 m: the ring height was 3.05 m) and the ball was horizontally cloth to the ring (less than 0.5 m: the ring diameter was 0.45 m), the ball was detected as a shot. (3) If the ball was neither holding nor shot, the ball was passing because the trial was ended when a defender held the ball. We analysed the duration before an attacker’s shot or ball loss. When an attacker gained the ball after a shot (i.e., an offensive rebound), if the ball was brought out of the remote place from the ring (more than 5 m), we analysed the attack from the time to the shot.

**Support Vector Machine (SVM).** For the classification problem in this study, soft margin SVM with Gaussian-kernel was employed. SVM can separate data of two categories with constructing a hyper-plane to maximise the margin between positive and negative instances. The nearest data point from the hyper-plane is called support vectors. In SVM, this discriminant function is used:

(1)

where is the feature vector, is a weight vector, is a scalar and represents the following sign function:

 (2)

This means that SVM creates a set of feature vectors to separate two sets by a hyper-plane. In the training set, is modulated and defined from training feature vector and correct label . is required to set to separate two sets with best quality. Therefore, is defined as the shortest distance between the separating hyper-plane and the vector closest to the hyper-plane, i.e., the ‘margin’. After all, we can replaces the problem to find a hyper-plane as the problem to find a parameter **.** Thus, the objective function is expressed as follows:

 (3)

under the limiting condition where is the ordinal number:

 (4)

The soft margin SVM tolerates a degree of error such that the method of SVM is applied to a nonlinear distribution. The error is given by , where is a parameter, so that the object function changes where is a penalty parameter :

(5)

under the following limiting condition:

 (6)

In addition, Gaussian-kernel () makes the nonlinear transformation to help linear separation such that a separation can be formed even if it can not separate linearly:

 (7)

**Classification with one-against-one SVM.** We used the one-against-one SVM to classify the detailed types of off-ball screen-plays. The one-against-one method constructed *k*(*k*-1)/2 SVM models, where *k* is the number of classes. To distinguish the th class and th class, in the th class is labelled positive, the th class is labelled negative and all other classes are ignored. Here, creating feature vectors and training each SVM is the same as a conventional SVM. The output of a classifier given as is the confidence of the binary classifier discriminating classes and in favour of the former class. The confidence of the classifier for the latter is computed by if the classifier did not provide it (the class with the largest confidence is the output class of a classifier).

After classification, each SVM votes for the predicted class. The votes for each class are counted, and the class with the greatest number of votes is regarded as predicted class, i.e.,

 (8)

where is a vote for the classifier to distinguish the th class and the th class and is 1 if and 0 otherwise.