

On-line Detection of Rule Violations in Table Soccer

Armin Hornung and Dapeng Zhang

Albert-Ludwigs-Universität Freiburg, Department of Computer Science
Georges-Köhler-Allee 052, 79110 Freiburg, Germany
{hornunga, zhangd}@informatik.uni-freiburg.de

Abstract. In table soccer, humans can not always thoroughly observe fast actions like rod spins and kicks. However, this is necessary in order to detect rule violations for example for tournament play. We describe an automatic system using sensors on a regular soccer table to detect rule violations in realtime. Naive Bayes is used for kick classification, the parameters are trained using supervised learning. In the on-line experiments, rule violations were detected at a higher rate than by the human players. The implementation proved its usefulness by being used by humans in real games and sets a basis for future research using probability models in table soccer.

1 Introduction

Table soccer – also known as “*Foosball*” – is a popular game, often played in pubs or other social contexts. Two teams of up to two players compete at scoring goals by controlling and kicking a ball on a game table with playing figures attached to rods. Playing table soccer requires skill at controlling the rods and quick reactions, because it is a very fast game. The ball can be kicked by rotating the rods, and professional players can shoot the ball across the table within a few hundred milliseconds.

Currently, the sport is also becoming more and more popular on a tournament level. There are various local and national leagues, and internationally, the ITSF¹ is organizing championships with a set of unified rules.

Because of the high speed of table soccer, a human is not always able to thoroughly observe all actions in the game, such as the fast turning of a kicking rod. However, this is necessary to make objective decisions about rule violations.

We present an approach to automatically detect rule violations on-line, using high frequency sensor data to classify and evaluate game situations. This helps humans playing table soccer, for example when practising for tournament games.

Most rules covering game mechanics depend on the detection of a playing figure kicking the ball. Thus, we concentrate on this key element, and detect violations of one of the most important rules: Rods may not be rotated by more than 360 degrees before or after ball contact, the angles before and after ball

¹ International Table Soccer Federation, <http://www.table-soccer.org/>

contact are not added. This prevents players from spinning the rods in an uncontrolled manner when kicking. In this paper, the coherent rotation movement of a rod is referred to as a *spin*.

The sequential sensor data for the game rods and ball is segmented, and game situations are classified as kicks. Because the sensor data is noisy, a probabilistic model for kick detection is needed. We use supervised learning to train a naive Bayes classifier for continuous variables.

The paper is organized as follows. We will first describe related work in the domain of table soccer and probability models, while deriving the theoretical background in Section 2. The methods used for segmentation and classification are described in Section 3, followed by the experimental setup and results in Section 4. Finally, we will conclude our work in Section 5.

2 Related Work

2.1 Robotic Table Soccer

The first work on a robot playing table soccer was published by Weigel and Nebel [1] in 2002. A regular soccer table was equipped with control units to manipulate the rods. The state was perceived by a camera mounted over the table. Analyzing the game state and controlling the actuators with a regular PC, *KiRo* (“Kicker Robot”) won most games against amateur players. In 2005, the first commercial product based on KiRo became available under the name “*Star-Kick*”, posing a challenge even to advanced players [2].

The current implementations of KiRo and Star-Kick do not take the enemy figures into consideration, the world model includes just the positions of the ball and the controlled figures. Future improvements could be to play on a professional level using a more advanced ball control, avoiding the enemy figures and planning actions such as passes.

For research in this direction and the work presented in this paper, a regular game table was outfitted with sensors to record ball positions as well as rod positions and angles [3].

2.2 Probability models

A common approach to model dependencies between random variables is a *Bayesian network* in the form of a directed graph [4]. Nodes represent random variables or classes, while directed edges denote dependencies. In a classification problem, the network is used to compute the posterior probabilities of all classes c_k given the observed values of the attributes: $P(c_k|x_1, \dots, x_n)$. The class with the highest probability is then assigned to the observation. Using *Bayes’ Theorem*, the posterior of a class c can be computed as

$$P(c|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n|c)P(c)}{P(x_1, \dots, x_n)}, \quad (1)$$

which flips the conditioning to make the distribution easily learnable.

We assume that all classes are equally probable, all attributes independent given the class and that the prior is independent of the class. This greatly simplifies computation to:

$$P(c|x_1, \dots, x_n) = \alpha \prod_{i=1}^n P(x_i|c) . \quad (2)$$

With α being a normalization constant, this is the naive Bayes classifier. Even though the “naive” assumptions seldom hold in reality, naive Bayes has been attested a good performance in many domains and can even be the optimal classifier with respect to the misclassification rate in some cases [5, 6].

In this work, observed values are not discrete but continuous. One common approach is to discretize the attributes [7]. However, we believe that discretization will lead to system degradation by discarding information like the probability for a kick. Instead, we model the random variables to be Gaussian-distributed and use the joint probability of the distributions for each variable.

Common extensions to naive Bayes include *tree-augmented naive Bayesian networks* [8], where additional dependencies between the attributes can be modeled. The cost is a higher algorithmic complexity. To cope with the high sampling frequency to observe games on-line, an efficient implementation is needed in our case. Our experiments revealed that the classification performance of naive Bayes is good enough for this application.

With continuous and Gaussian-distributed attributes in (2), it follows that:

$$P(c|x_1, \dots, x_n) = \alpha \prod_{i=1}^n \varphi_i(x_i) = \alpha \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}} . \quad (3)$$

Variance σ_i^2 and mean μ_i of each attribute i are determined with supervised learning for a known class label c . m attribute values $x_{i,1}, \dots, x_{i,m}$ are sampled for each attribute i and the unbiased estimators for mean and variance are used on these values.

3 Segmentation and Classification

3.1 Sensor Input

To cope with the high speed of the game, the sensor data is read and recorded at roughly 250 Hz using a standard PC, distributing the data stream over network for further evaluation on other machines.

The ball is located with a *Sick LMS 400* laser measurement system, scanning through the gap between table surface and feet of the playing figures. Rod positions are measured with optical distance sensors and rod angles are observed with magnetic rotary encoders. Overall, there is almost no additional friction on the rods, enabling an unhindered game play.

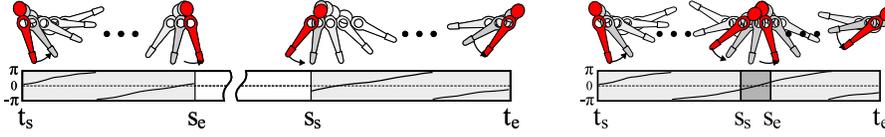


Fig. 1. The timestamps t_s, s_e, s_s and t_e defining one complete spin in a rotation covering an angle of more (left) and less than 4π (right).

To analyze the game state, one needs to segment the sequential sensor data and detect key events in it. The sensors introduce Gaussian-distributed noise on the measured signal.

Currently, three rods are observed by sensors, all in the same half of the table: blue attacker, red defender and red goalkeeper.

3.2 Rotation Segmentation

With respect to the rule violations described in Section 1, a rod spin is completely parametrized by four timestamps: t_s and t_e , the starting and end times of the whole spin, as well as s_e and s_s , the times when the first 2π part of the rotation ends and the time when the last 2π part starts (Fig. 1). These discrete timestamps need to be detected in the stream of angle measurements for each rod i .

To do so, the difference in between two successive rod angles α_i is observed as approximation of the derivative:

$$\Delta_i(t) = \alpha_i(t) - \alpha_i(t - 1) . \quad (4)$$

When crossing from 2π to 0, the values are adjusted accordingly. As soon as $|\Delta_i(t)| \geq \varepsilon$ for some small threshold ε , the start of a spin movement is detected, and it lasts until $|\Delta_i(t)| < \varepsilon$. To reduce the influence of noise, the signal is smoothed by using the running average over a window of size three.

As soon as $\alpha_i(t_s)$ is passed for the second time, the time s_e is detected. s_s depends on $\alpha_i(t_e)$, the angle at which the spin stops, and is detected by using a circular array or ring buffer indexed by angle. For additional robustness against noise, monotonicity in between start and end of the spin is enforced when storing timestamps in the circular array. Also for noise robustness, all bins in between storing two successive timestamps need to be emptied.

3.3 Kick Detection

Naive Bayes classifiers are used to detect kicks, by using the relation of ball and active figure as input. The active figure is the one closest to the ball on the rod that is within range of the ball. The two directions of a kick are distinguished by classifying two cases, forward and backward.

Input for each classifier are the continuous attributes x_1, x_2, x_3 , computed from coordinates of ball and active rod, and its angle. Figure 2 displays the

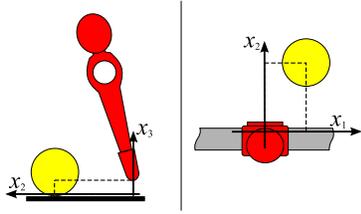


Fig. 2. Coordinates of ball relative to the active figure from the side (left) and top(right)

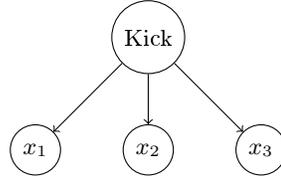


Fig. 3. Bayesian network of the kick classifier

coordinates relative to the playing figure, Fig. 3 displays the resulting Bayesian network. Note that x_3 is directly related to the vertical distance between ball and playing figure, because it is assumed that the ball always touches the table when kicking. When it is not on the surface, e.g. resulting from a fast kick, it cannot be observed by the laser anyways.

The probability can now be computed according to (3) with $n = 3$ variables. In the implementation the log-likelihood is used instead, turning the products into sums. This is computationally more stable because very small probabilities are avoided, while it is equivalent in terms of classification. Furthermore, the constant α is ignored for classification because it is assumed to be identical for all classes.

To detect violations of the rod-spinning rule efficiently, the log-likelihood for kicks is constantly recorded. As soon as a spin covering an angle of more than 2π is detected, the peak of the kick likelihood in the intervals $[s_e, t_e]$ and $[t_s, s_s]$ (see 3.2) is compared to an experimental threshold. Depending on the direction of the spin, the forward or backward probability is used.

4 Experiments

In the experiments, we first evaluate the supervised learning performance of the kick classifier. Then, the parameters for the kick model for on-line detection are trained. Finally, we test the performance of violation detection in real table soccer games on-line.

4.1 Supervised Learning Performance

To evaluate the learning performance, 50 recorded kick actions were randomly partitioned into a test set of size 10 and a training set of size 40. In each single action k , the instant when the playing figure touches the ball is selected by hand, and the variable state $(x_{1,k}, x_{2,k}, x_{3,k})$ is extracted.

The parameters of the kick model μ_i, σ_i for $i \in \{1, 2, 3\}$ are then learned incrementally, using $k = 1, \dots, 40$ samples of the training set as input. The performance on classifying the test set is evaluated for each step.

As performance measure V , the sum over the normalized probability of each test sample $x_{i,l}$ is used:

$$V(k) = \sum_{l=1}^{10} \gamma_k \prod_{i=1}^3 \varphi_{i,k}(x_{i,l}) . \quad (5)$$

The Gaussian distribution $\varphi_{i,k}$ is parametrized by mean and variance of variable x_i , using k samples of the training set. The normalization constant is determined by the maximum of the joint probability, reached at the mean:

$$\gamma_k = \prod_{i=1}^3 (\varphi_{i,k}(\mu_{i,k}))^{-1} . \quad (6)$$

This scales all Gaussian distributions in (5) to the range of $[0, 1]$. Otherwise, their probabilities would not be comparable, because a more general model creates a more shallow distribution.

The resulting learning curve is shown in Fig. 4. A first peak is reached after using only six samples for training, and the performance of the learned model stays stable after using 16 samples for training. A training set significantly larger than 20 samples only leads to small improvements.

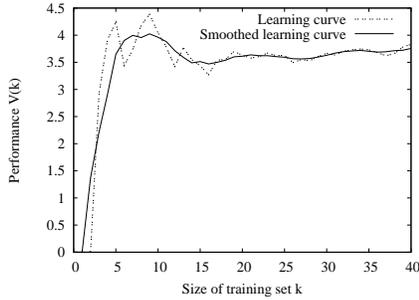


Fig. 4. Learning curve for 40 kick samples on a testing set of size 10. The smoothed curve displays the average over a running window of five values.

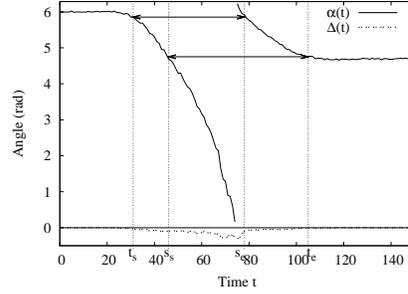


Fig. 5. Angles and first derivative for one spin, lasting about 0.3s. The detected times of t_s, s_e, s_s, t_e are marked.

4.2 Parameter Learning

To classify kicks in running games, only the model for a forward kick is learned. The parameters for a back kick are obtained by negating μ_2 .

To train the parameters of the kick model, the variable state (x_1, x_2, x_3) is sampled from several configurations with the ball in front of a playing figure, varying positions, angles and playing figures.

Table 1. Parameters obtained from training by sampling kick positions

$i =$	1	2	3
Mean μ_i	-1.37	23.54	13.02
Variance σ_i^2	30.30	7.09	3.21

Table 1 displays the parameters trained through the sampling process, using 27 samples. As expected, the peak is centered in front of the playing figure. The rectangular footprint of the playing figure results in a larger variance in x_1 than in x_2 .

4.3 Spin Detection

Spins covering angles of more than 2π and the classifying timestamps t_s , s_e , s_s and t_e are successfully detected. Figure 5 exemplarily shows the sensor recording of one spin movement starting at $t_s = 31$ and ending at $t_e = 105$, with the detected timestamps marked. Note the noisy signal for $\alpha(t)$, and that the whole spin from the angle $\alpha(t_s) \approx 5.85$ to $\alpha(t_e) \approx 4.75$ (rad) lasts only 74 sensor ticks, which is about 0.3s.

4.4 Detecting violations on-line

To evaluate the detection rate in real games, test subjects not familiar with the system played several games on the table. The position of the players on the table (attacker or defender for the blue or red team) were exchanged regularly. Two players rated themselves as “good”, two “average” and the remaining seven as “amateurs”. In total, 9 games were recorded for about 1 hour and 45 minutes, silently logging the detected rule violations. Before the games, the rod spinning rule was explained to the players, and they were asked to evaluate whether they think they violated the rule after each shot. Afterwards, the recorded sensor logs were manually inspected in slow-motion for rule violations, and compared to the detections of the system and the players.

All in all, there were 42 rod spins of more than 360 degrees on the observed three rods, of which 19 were illegal kicks. The players themselves were aware of only two of them (10.52%), while the system detected 17 violations correctly (89.47%). Two violations were missed by the system, and one false positive detected. When a rod is spun and misses the ball closely instead of kicking it, there is a chance that a rod spin violation is detected. The noise on the rod and ball data might add up so that the ball is falsely located as being touched by the rotating figure, resulting in the false positive detection. False negatives occur for similar reasons.

The referee system runs efficiently on a standard PC with a 2.66 GHz Pentium 4 CPU and 1 GB RAM, running SuSE Linux 10.1. The application uses just a small fraction of the available CPU power. Most of the CPU power is used for a 3D display of the soccer table in the user interface, which shows the live

representation of the soccer table and slow-motion replays of rule violations. The distributed implementation of the system allows the display to be easily outsourced to a dedicated machine, leaving more processing power for example to detect additional rules.

5 Conclusion

In this work, we presented an approach to automatically detect rule violations in table soccer games. A naive Bayes classifier is trained off-line to detect kicks, using the relation of ball and closest playing figure as input.

The classifier demonstrated a good performance in the on-line classification experiments, detecting 89.47% of all rule violations, while the human players only detected 10.52%. The efficient implementation of naive Bayes enables the system to run effortlessly on a standard PC, evaluating the high-frequency sensor data on-line during a running game. All this demonstrates the usefulness of our implementation. Additional robustness on the classification method can be achieved in future work by taking the ball movement into account for kick detection.

Finally, future research on table soccer can benefit from the classification method described here, such as learning by imitation or game analysis. The classifier could be used to detect various relations between ball and playing figures.

References

1. Weigel, T., Nebel, B.: Kiro – an autonomous table soccer player. In: RoboCup 2002, Springer-Verlag (2002) 384–392
2. Weigel, T., Nebel, B.: Kiro – a table soccer robot ready for the market. In: ICRA, Spain (2005) 4277–4282
3. Zhang, D., Nebel, B.: Recording and segmenting table soccer games – initial results. In: ISSS'07, Keio University, Tokyo, Japan (2007) 193–195
4. Pearl, J.: Bayesian networks: A model of self-activated memory for evidential reasoning. In: Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine. (August 1985) 329–334
5. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* **29**(2-3) (1997) 103–130
6. Zhang, H.: The optimality of naive Bayes. In Barr, V., Markov, Z., eds.: FLAIRS Conference, AAAI Press (2004)
7. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: International Conference on Machine Learning. (1995) 194–202
8. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* **29**(2-3) (1997) 131–163