Comparing Fuzzy Naive Bayes and Gaussian Naive Bayes for Decision Making in RoboCup 3D

Carlos Bustamante, Leonardo Garrido, and Rogelio Soto

Centro de Sistemas Inteligentes
Tecnológico de Monterrey
Monterrey, N.L. 64849, México.
{cfbh, leonardo.garrido, rsoto}@itesm.mx

Abstract. Learning and making decisions in a complex uncertain multiagent environment like RoboCup Soccer Simulation 3D is a non-trivial task. In this paper, a probabilistic approach to handle such uncertainty in RoboCup 3D is proposed, specifically a Naive Bayes classifier. Although its conditional independence assumption is not always accomplished, it has proved to be successful in a whole range of applications. Typically, the Naive Bayes model assumes discrete attributes, but in RoboCup 3D the attributes are continuous. In literature, Naive Bayes has been adapted to handle continuous attributes mainly using Gaussian distributions or discretizing the domain, both of which present certain disadvantages. In the former, the probability density of attributes is not always well-fitted by a normal distribution. In the latter, there can be loss of information. Instead of discretizing, the use of a Fuzzy Naive Bayes classifier is proposed in which attributes do not take a single value, but a set of values with a certain membership degree. Gaussian and Fuzzy Naive Bayes classifiers are implemented for the pass evaluation skill of 3D agents. The classifiers are trained with different number of training examples and different number of attributes. Each generated classifier is tested in a scenario with three teammates and four opponents. Additionally, Gaussian and Fuzzy approaches are compared versus a random pass selector. Finally, it is shown that the Fuzzy Naive Bayes approach offers very promising results in the RoboCup 3D domain.

1 Introduction

RoboCup simulation is an excelent test-bed for machine learning algorithms. It presents a multi-agent cooperative and adversarial scenario in a partially observable, episodic, continuous and non-deterministic noisy environment.

Given such uncertainty, classical logic-based approaches fail to achieve a high performance. Thus, a probabilistic method is ideal for dealing with this kind of environment.

The most simple probabilistic approach is the Naive Bayesian classification [1] which has proven to be successful in many applications [2] in spite of the not always fulfilled conditional independence assumption of the attributes given the class. If we wish to use this classifier in the RoboCup simulation domain, we confront two main issues.

First, the classical Naive Bayes classifier assumes that the attributes are discrete, but in RoboCup simulation the attributes are in the range of real numbers and thus are continuous. Second, the classifier must lead to a fast decision process because the soccer simulator demands almost real-time decisions with low thinking times for the sense-think-act cyle of the agents.

In literature, continuous attributes are handled using conditional Gaussian distributions for each attribute likelihood given the class. Other approach is to discretize by crisp partitioning the domain of the attributes, but this can lead to loss of information.

Instead of discretizing, we overcome the issues using a fuzzy extension namely Fuzzy Naive Bayesian classifier in the following way: the continuous attributes are fuzzified and combined with probabilities of the naive bayes model in a straight easy way. The formulas used in the fuzzy extension resemble the original naive bayes equations, so the classification process is still fast and reliable plus providing an incremental learning mechanism.

The Fuzzy Naive Bayesian classifier is implemented in a RoboCup simulation 3D team for decision making. We test it specifically to evaluate the best pass receiver in a given situation. In the next sections, we first explain the Fuzzy Naive Bayes model and the Gaussian model, then we describe our empirical scenarios for the pass evaluation skill. After that, results of our experiments and some conclusions are shown.

2 Naive Bayes and the fuzzy extension

The Naive Bayes classifier is a simple bayesian network with one root node that represents the class and n leaf nodes that represent the attributes. Let C be a class label with k possible values, and $X_1 ... X_n$ be a set of attributes or features of the environment with a finite domain $D(X_i)$ where $i = 1..n$. The classifier is given by the combination of the bayesian probabilistic model with a maximum a posteriori (MAP) rule, also called discriminant function [3]. The Naive Bayes classifier is defined as follows

$$NBayes(a) = argmax_{c \in C} P(c) \prod_{i=1}^{n} P(x_i|c) \tag{1}$$

where $a = \{X_1 = x_1, ..., X_n = x_n\}$ is a complete assignation of attributes, i.e. a new example to be classified, x_i is a short for $X_i = x_i$ and c is a short for $C = c$. The equation assumes conditional independence between attributes.

To deal with continuous variables, the domain of attributes can be crisp partitioned, but that could cause a loss of information [4]. We use a better method proposed in [5], namely a Fuzzy Bayesian classifier, a hybrid approach in which attributes are fuzzified before classification.

In this approach, the degrees of truth are treated as probabilities such that $P(x_i|a) = \mu_{x_i}$ and $P(c|a) = \mu_c$. Although degrees of truth represent membership values of classes rather than probabilities, it allows a natural extension of the classical Naive Bayes equation, using the Bayes' rule and assuming independence among attributes

$$P(c|a) = \sum_{x_1 \in X_1,..x_n \in X_n} P(c|x_1..x_n)P(x_1|a)..P(x_n|a) \tag{2}$$

$$P(c|a) = \sum_{x_1 \in X_1,..x_n \in X_n} \frac{P(x_1|c)..P(x_n|c)P(c)}{P(x_1)..P(x_n)} \mu_{x_1}..\mu_{x_n} \tag{3}$$

The Fuzzy Naive Bayesian classifier is defined from (3) as follows

$$FNBayes(a) = argmax_{c \in C} P(c) \sum_{x_{1j} \in X_1} \frac{P(x_{1j}|c)}{P(x_{1j})} \mu_{x_{1j}}.. \sum_{x_{nj} \in X_n} \frac{P(x_{nj}|c)}{P(x_{nj})} \mu_{x_{nj}} \tag{4}$$

where $j = 1..D(X_i)$ and $\mu_{x_{ij}} \in [0,1]$ denotes a membership function or degree of truth of attribute value $x_{ij} \in X_i$ in a new example a. All degrees of truth must be normalized such that $\sum_{x_{ij} \in X_i} \mu_{x_{ij}} = 1$ for all attributes $i = 1..n$.

The probabilities required by the fuzzy model can be calculated similarly to classical Naive Bayes (1)

$$P(C = c) = \frac{(\sum_{e \in L} \mu_c^e) + 1}{|L| + |D(C)|} \tag{5}$$

$$P(X_i = x_i) = \frac{(\sum_{e \in L} \mu_{x_i}^e) + 1}{|L| + |D(X_i)|} \tag{6}$$

$$P(X_i = x_i|C = c) = \frac{(\sum_{e \in L} \mu_{x_i}^e \mu_c^e) + 1}{(\sum_{e \in L} \mu_c^e) + |D(X_i)|} \tag{7}$$

where Laplace-correction [6] is applied to smooth calculations avoiding extreme values obtained with small training sets. Here L is the set of all training examples e, where $e = \{X_1 = x_1, ..., X_n = x_n, C = c\}$, $|L|$ refers to the number of examples $e \in L$, $\mu_c^e \in [0,1]$ denotes the degree of truth of $c \in C$ in a example $e \in L$, and $\mu_{x_i}^e \in [0,1]$ is the membership of attribute $x_i \in X_i$ in such example. Same as in (4), all degrees of truth must be normalized such that $\sum_{c \in C} \mu_c^e = 1$ and $\sum_{x_i \in X_i} \mu_{x_i}^e = 1$.

3 Gaussian Naive Bayes

One typical way to handle continuous attributes in the Naive Bayes classification is to use Gaussian distributions [7] to represent the likelihoods of the features conditioned on the classes. Thus each attribute is defined by a Gaussian probability density function (PDF) as

$$X_i \sim N(\mu, \sigma^2) \tag{8}$$

The Gaussian PDF has the shape of a bell and is defined by the following equation

$$N(\mu, \sigma^2)(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{9}$$

where μ is the mean and σ^2 is the variance. In Naive Bayes, the parameters needed are in the order of $O(nk)$, where n is the number of attributes and k is the number of classes. Specifically we need to define a normal distribution $P(Xi|C) \sim N(\mu, \sigma^2)$ for each continuous attribute. The parameters of such normal distributions can be obtained with

$$\mu_{X_i|C=c} = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i \tag{10}$$

$$\sigma^2_{X_i|C=c} = \frac{1}{N_c} \sum_{i=1}^{N_c} {x_i}^2 - \mu^2 \tag{11}$$

where N_c is the number of examples where $C = c$ and N is the number of total examples used for training. Calculating $P(C = c)$ for all classes is easy using relative frequencies such that

$$P(C = c) = \frac{N_c}{N} \tag{12}$$

4 Empirical scenarios

Selecting a good scenario for training the classifiers is not trivial. In simulated soccer there is a large set of possible scenarios for a given skill.

For the purposes of this paper, we chose the pass evaluation skill as our test-bed for the training of both classifiers. One of the reasons why we selected this skill was that passing is a fundamental characteristic of an agent that aims to play a soccer game. Specifically, deciding what teammate is the best receiver in a given situation could lead to better chances to score later in the game.

Some may argue that passing not only involves the ability to select an optimal receiver, but the ability of the receiver to predict the pass action and anticipate it in some way. Additionally, it could be said that the teammate with best chances to receive a ball and control it is not always the best receiver.

Consider the situation when the ball is controled by a midfielder, the probability of a successful pass is just a little bigger for a defender than for a forward, just evaluating the pass would select the defender as the best passing option. However, it is easy to see that the forward has more chances to score generally, so the **utility** of passing to the forward is greater than the utility to passing to a defender. These arguments may be true, but that concerns a higher layer which some people have called pass selection [8] and involves the calculus of expected utilities.

This paper is focused on the pass evaluation only, because our main goal is to compare different strategies to calculate the probabilites for selecting a receiver. In [9] we introduced the use of a Fuzzy Naive Bayes classifier for the pass evaluation skill. We tested the classifier, trained with 1000 examples, in 300 episodes of a test scenario. The results were promising: 76% of the 300 passes were successful. However, we did not compared the Fuzzy Naive Bayes classifier to another approach. In this paper we make the comparison between both classifiers in the same scenarios to evaluate the performance of the Fuzzy Naive Bayes in a more tangible way.

The scenario we used to obtain the training set is explained below. One agent (**passer**) is placed in the center of the field with the ball at a distance of $d_{AB} \in [kickrange, 2]$, where *kickrange* is the minimum kicking radial distance between the agent and the ball stated in the soccer server. Another agent (**teammate**) is placed near the ball at a distance $d_{TB} \in [2, 20]$. One last agent (**opponent**) is placed similarly, with a distance $d_{OB} \in [2, 20]$ from the ball. The angle between the teammate and the opponent from the ball's view point must be $\alpha \in [0, \frac{\pi}{6}]$. A graphical representation of this scenario is shown in figure 1.

This scenario is a modification of the one proposed in [10] and used in simulation 2D. We added the alignment angle and distance to the ball because in 3D soccer, as agents are spheres, the force for kicking the ball is applied radially, thus the agent has to position correctly around the ball before kicking and of course, losing some time.

The features extracted from the scenario in each episode are:

– Distance to the ball d_{AB}
– Distance to teammate d_{AT}
– Distance to opponent d_{AO}
– Alignment angle $\theta \in [0, \pi]$
– Angle between teammate and opponent α

The behaviors of each agent during the episode are as follows:

– The passer agent aligns with the ball to pass it to its teammate.
– The teammate and the opponent try to intercept the pass.

Once the teammate touched the ball, the episode is labeled as *SUCCESS*. If the opponent touches the ball first, episode is labeled as *MISS*.

In the case of the Fuzzy Naive Bayes classifier, aside of obtaining the probabilities of the bayesian model, we have to establish the fuzzy sets for each variable.

Fuzzy sets represent linguistic values and are mathematically expressed with membership degree functions. We defined the fuzzy sets for each variable heuristically. The sets chosen for distance to the ball d_{AB}, distance to teammate d_{AT} and distance to opponent d_{AO} variables are {*short, medium, long*}, and for θ

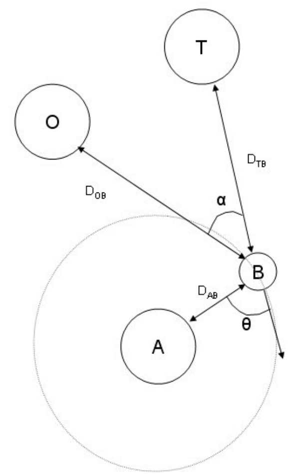

Fig. 1. Training scenario for supervised learning of parameters of each classifier. Three agents are involved: a passer agent (A), a receiver teammate (T) and an opponent (O). The ball is marked as (B). Features considered are: distance to ball d_{AB}, distance to teammate d_{TB}, distance to opponent d_{OB}, alignment angle (θ) and angle bewteen teammate and opponent from the ball's view point (α).

and α variables are {*closed, medium, wide*}. A graphical representation of each fuzzy variable is shown in figure 2.

The probabilities for the Fuzzy Naive Bayes model are calculated using equations (5), (6) and (7). The learning process is described as follows:

- Read the training set one example at a time.
- For each attribute of each example, calculate the normalized membership degrees of each fuzzy set.
- Store the sums $\sum_{e \in L} \mu_c^e$, $\sum_{e \in L} \mu_{x_i}^e$ and $\sum_{e \in L} \mu_{x_i}^e \mu_c^e$. Note that μ_c^e is either 1 or 0 because the class attribute is discrete. Additionally, store the total numer of examples L and the domain size of the class $|D(C)|$ and attributes $|D(X_i)|$.
- Compute all the probabilities $P(C = c)$, $P(X_i = x_i)$ and $P(X_i = x_i|C = c)$.
- When a new example arrives, use the calculated probabilities to classify it with equation 4.

5 Experimental Results

We ran two main tests, one for evaluating the performance of both classifiers in a test set and the other for evaluating the efficiency in a simulated soccer scenario.

We used a test set of size 500. We can see the performance of both classifiers with different number of variables in figures 3 and 4. We ordered the variables

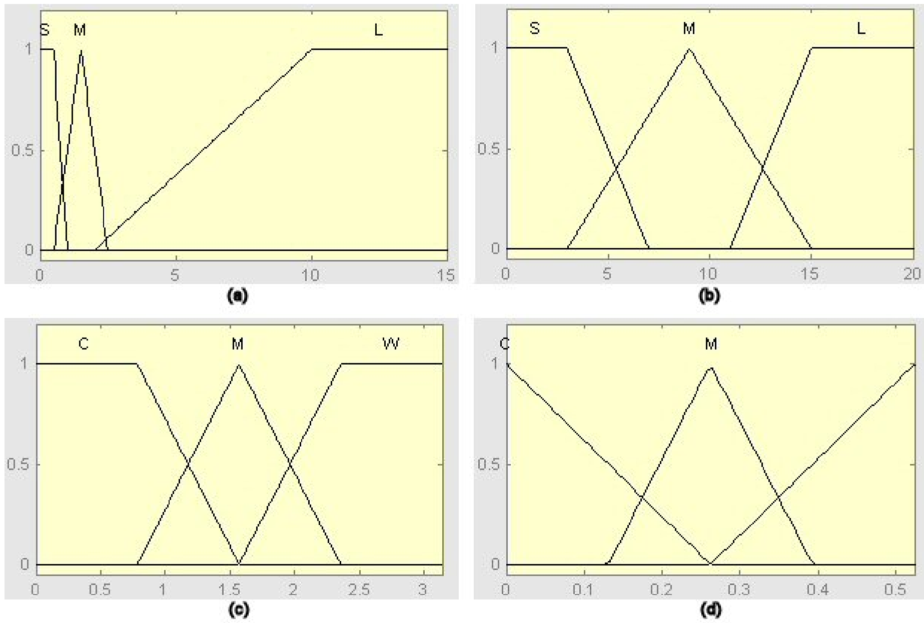

Fig. 2. Fuzzy Sets for each Fuzzy Variable. (a) Distance to the ball d_{AB}, (b) Distance to teammate d_{AT} and distance to opponent d_{AO}, (c) Alignment Angle θ and (d) Angle between teammate and opponent α.

heuristically according to their relative importance, considering that the distance to teammate is the most important and the angle θ is the less important:

– Distance to teammate d_{AT}
– Distance to opponent d_{AO}
– Angle between teammate and opponent α
– Distance to the ball d_{AB}
– Alignment angle $\theta \in [0, \pi]$

For evaluating the efficiency in the domain of interest, we created a simulated-soccer test-scenario shown in figure 5. The ball is placed at $(x = -20, y = 0)$ and the agent is placed at $(x \in [-22, -18], y \in [kickrange, 2])$. After that, three teammates and four opponents are placed randomly at $(x \in [-30, -10], y \in [10, 30])$.

The passer uses a classifier to choose the best receiver teammate, i.e. the teammate with better chances to intercept the pass successfully. The passer uses the classifier evaluating all 1 vs. 1 competitions between each teammate and each opponent (because the classifier was trained this way). Then it selects the teammate with the maximum probabilty of success given its worst probabilty in all its 1 vs. 1 competitions, formally

$$Receiver = argmax_{t \in T}\ argmin_{o \in O}\ P(SUCCESS_{to}) \qquad (13)$$

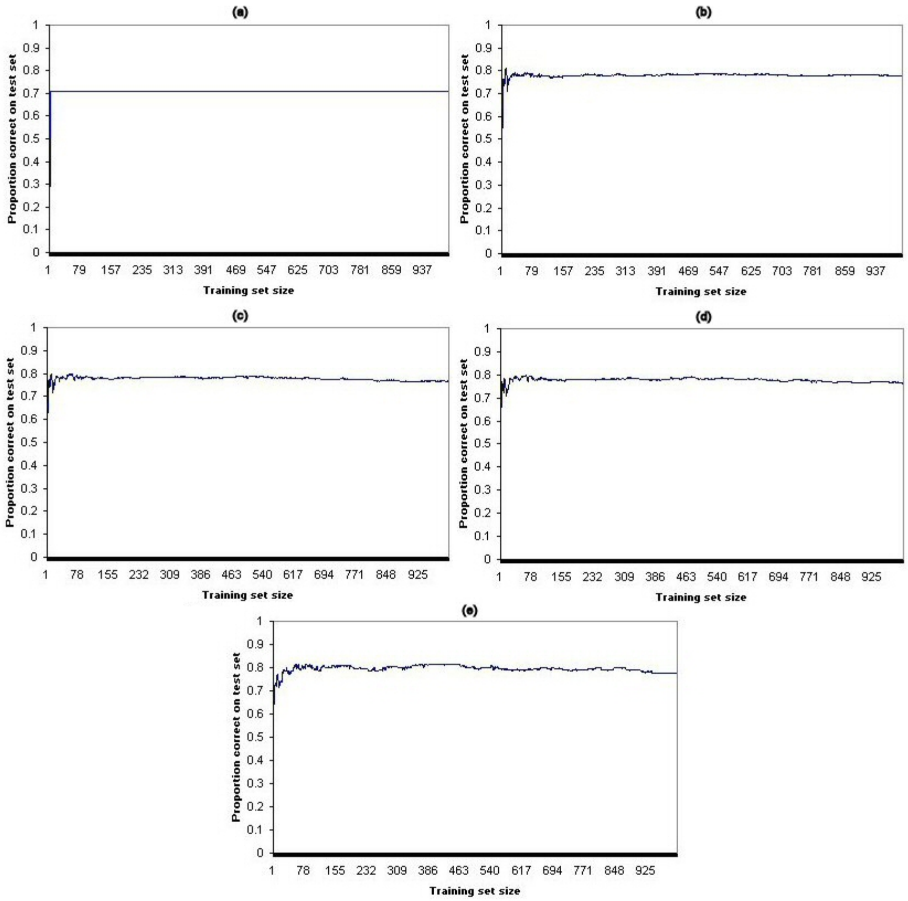

Fig. 3. Performance of the Fuzzy Naive Bayes classifier using (a) one attribute, (b) two attributes, (c) three attributes, (d) four attributes and (e) five attributes.

being T the set of all teammates, O the set of opponents and $P(SUCCESS_{to})$ is the probability of success of the competition between teammate $t \in T$ and opponent $o \in O$. The episode is a $SUCCESS$ if the teammate intercepts the ball before any opponent does, otherwise, the episode is labeled *MISS*. Indeed, this is the way to use the classifier in a real game.

In table 1, we summarize the success rates of Fuzzy Naive Bayes, the Gaussian Naive Bayes and additionally, a random strategy. We classified 500 episodes with each classifier trained with different number of variables. The maximum percentage of success was achieved using 5 variables for both the fuzzy and the gaussian approaches.

As we can see in table 1, both the Fuzzy Naive Bayes classifier and the Gaussian Bayes classifier outperform the random strategy. But the difference

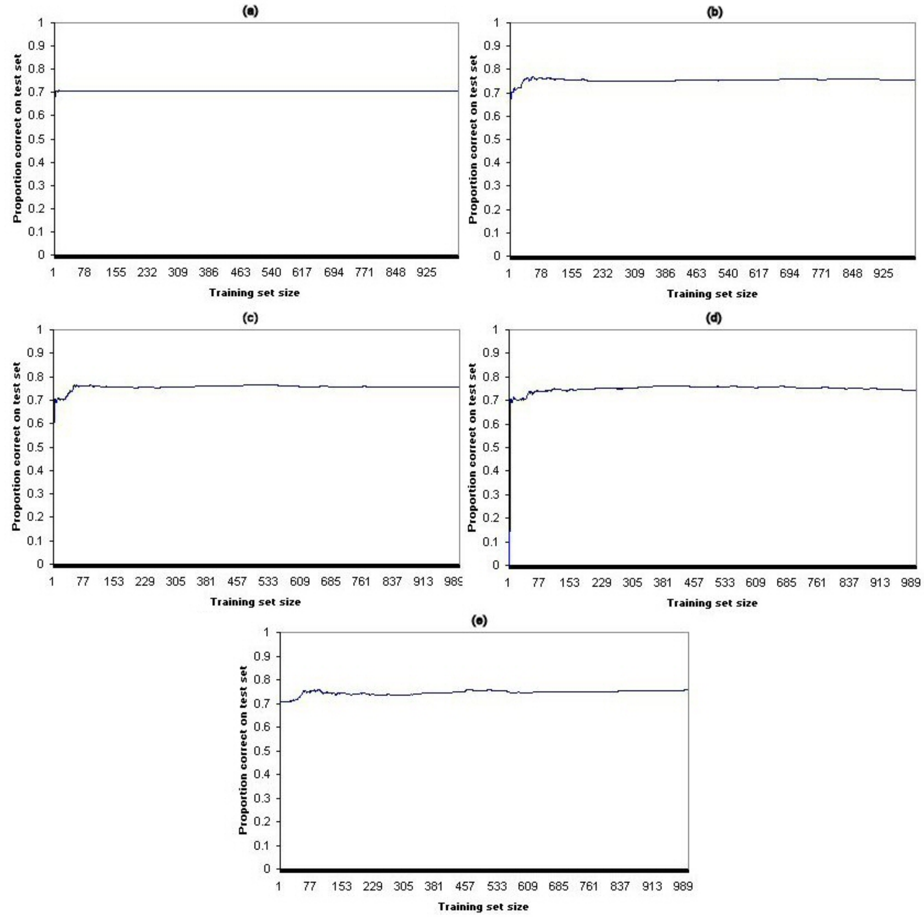

Fig. 4. Performance of the Gaussian Naive Bayes classifier using (a) one attribute, (b) two attributes, (c) three attributes, (d) four attributes and (e) five attributes.

between the Fuzzy Bayes and the Gaussian Bayes approaches is indiscernible. The Fuzzy Naive Bayes classifier is just 1.2% more accurate than the Gaussian Bayes classifier.

However, recall that fuzzy variables and fuzzy sets for each variable were chosen heuristically. The result seems more promising from this point of view, because it leaves an open path for researching the use of better variables and more accurate sets to increase the performance of the Fuzzy Naive Bayes classifier.

6 Conclusions

In this paper, we compared the Gaussian Naive Bayes classifier with a Fuzzy Naive Bayes classifier for making decisions. We focused on the pass evaluation

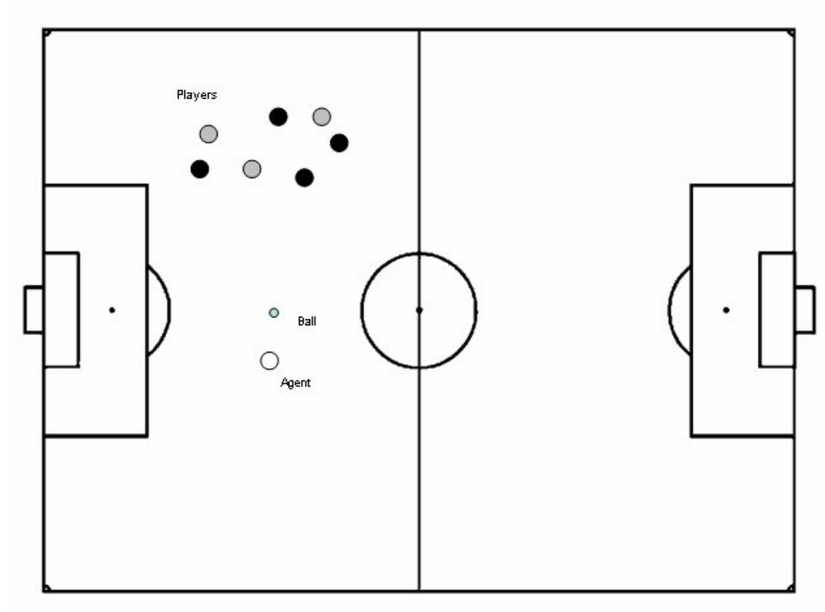

Fig. 5. Test scenario for the pass evaluation skill. Four opponent agents (black circles) and three teammates (gray circles) are placed randomly in a certain area. The passer (white circle) and the ball (little circle) are placed a few meters away. The passer chooses the teammate with the best chances to intercept the ball using the classifier.

Table 1. Percentage of successful passes for 500 episodes using our test scenario.

Class	Fuzzy Naive Bayes	Gaussian Naive Bayes	Random Strategy
SUCCESS	80.8	79.6	56.6
MISS	19.2	20.4	43.4

skill in the Robocup simulation 3D domain. Robocup simulation offers an excelent testbed for probabilistic learning algorithms because of the uncertain and noisy sensor data and unknown opponent model.

The Naive Bayes classifier has been successfully used in a whole range of applications although its conditional independence assumption is not always met. Naive Bayes attributes are usually discrete, but in RoboCup Simulation 3D attributes are continuous, thus we tried both Gaussian and Fuzzy extensions to classical Naive Bayes for handling those continuous features.

We trained both classifiers with different number of variables and tested them on the scenario proposed in figure 5. We obtained 80.8% of successful passes using the Fuzzy Naive Bayes approach trained with 5 variables. Stone [11] used a Decision Tree for pass evaluation in a similar test scenario and just 65% of all passes where successful.

As we can see in table 1, the Fuzzy Naive Bayes classifier outperforms the random strategy, but is just a little better than the Gaussian Naive Bayes approach. In fact, at a first glance, the results between the Fuzzy Naive Bayes classifier and the Gaussian Naive Bayes seem indiscernible. Nevertheless, the result obtained with the Fuzzy Naive Bayes classifier is very good considering that the variables and the fuzzy sets were chosen heuristically.

There is a lot of future work to do in this area. First of all, we have the hypothesis that if we define better sets and different variables we can get better results. Possibily, combining the classifier with a decision tree algorithm, we can prune the useless variables from our list of variables related to the pass skill. Also we want to test the classifier for other skills, like dribble and shoot. We plan to implement fuzzy k-means clustering for obtaining the fuzzy sets automatically from data. We also want to try other probability distributions aside from the Gaussian approach.

Acknowledgements

This work was supported in part by the ITESM research grant CAT-011 on distributed knowledge and intelligent agents technologies.

References

1. Langley P., Iba, W., Thompson, K.: An Analysis of Bayesian Classifiers. Proc. 10th Nat. Conf. on Artificial Intelligence, AAAI Press and MIT Press, USA (1992) 223-228
2. Lewis, D: Naive Bayes at forty: The independence assumption in information retrieval. In Proceedings of European Conference on Machine Learning, (1998) 4–15
3. Rish, I.: An empirical study of the naive bayes classifier. In Proceedings of IJCAI-01 workshop on Empirical Methods in AI, (2001) 41–46
4. Friedman, N., Goldszmidt, M.: Discretization of continuous attributes while learning Bayesian networks. In L. Saitta, editor, Proceedings of 13-th International Conference on Machine Learning, (1996) 157–165
5. Störr, Hans-Peter: A compact fuzzy extension of the Naive Bayesian classification algorithm. In Proceedings InTech/VJFuzzy, (2002) 172–177
6. Zadrozny, B., Elkan, E.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In Proceedings of 18th International Conf. on Machine Learning, (2001) 609–616
7. Mitchell, T: Machine Learning. McGraw-Hill, 1997.
8. Stone, P., Veloso, M.: Layered Learning. In Proceedings of 11th European Conference on Machine Learning, (2000) 369–381
9. Bustamante, C., Garrido, L., Soto, R.: Fuzzy Naive Bayesian Classification in RoboSoccer 3D: A hybrid approach to decision making. In Proceedings of RoboCup International Symposium, (2006).
10. Buck, S., Riedmiller, M.: Learning Situation Dependent Success Rates Of Actions In A RoboCup Scenario. Pacific Rim International Conference on Artificial Intelligence, (2000) 809
11. Stone, P.: Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer. MIT Press, 2000.