Comparison of Classification Methods for Golf Putting Performance Analysis

J. Miguel A. Luz, Micael S. Couceiro
RoboCorp, Department of Electrical Engineering (DEE)
Engineering Institute of Coimbra (ISEC)
Coimbra, Portugal
e-mail: {miguel.luz, micael }@isec.pt

David Portugal, Rui P. Rocha, Hélder Araújo
Institute of Systems and Robotics (ISR)
University of Coimbra (FCTUC)
Coimbra, Portugal
e-mail: { davidbsp, rprocha, helder}@isr.uc.pt

Gonçalo Dias
RoboCorp, Faculty of Sport Sciences and Physical Education, University of Coimbra (FCDEF)
Coimbra, Portugal
e-mail: goncalodias@fcdef.uc.pt

***Abstract*— This paper presents a comparative case study on the classification accuracy between 5 methods for golf putting performance analysis. In a previous work, a digital camera was used to capture 30 trials of 6 expert golf players. The detection of the horizontal position of the golf club was performed using a computer vision technique followed by the estimation algorithm Darwinian Particle Swarm Optimization (*DPSO*) in order to obtain a kinematical model of each trial. In this paper, the estimated parameters of the models are used as sample and training data of five classification algorithms: 1) Linear Discriminant Analysis (*LDA*); 2) Quadratic Discriminant Analysis (*QDA*); 3) Naïve Bayes with Normal (Gaussian) distribution (*NV*); 4) Naïve Bayes with Kernel Smoothing Density Estimate (*NVK*) and 5) Least Squares Support Vector Machines with Radial Basis Function Kernel (*LS-SVM*). The 5 classification methods are then compared through the analysis of the confusion matrix and the area under the Receiver Operating Characteristic curve (*AUC*).**

Keywords: golf; putting; classification; evaluation.

I. INTRODUCTION

The putt is a motor skill that combines art and science [1], representing about 43% of the strokes in a Golf game [2]. Coordination and control of this movement is an individual process that is different for each subject depending on his profile and characteristics [3]. Several studies [4-7] analyzed putting based on process measurements of motor execution, *i.e.*, movement action parameters like position, velocity and acceleration (Figure 1). The same researchers assume that the aspects of stability and variability in the execution of this movement, observed at expert level players and beginners alike, may have significant differences in motor performance. Like other motor skills, the intra and inter-individual performance resulting from the execution of the putt is a "fingerprint" (*i.e.*, signature) that is unique to each individual [8-11]. However, only a few studies analyzing process variables such as the position, velocity or acceleration (linear or angular) in the golf club, more specifically the putter, during putt execution [6] have been performed and, researches using automatic tracking of the putter's trajectory on the green or in laboratory are not known.

A previous work [12] presented the experimental design and methodological aspects in the analysis of the effects of variability in golf putting performance of expert subjects based on previously detected and estimated data of the horizontal position of the putter during the execution. The process variables, such as the trajectory function were obtained by estimation of a sinusoidal model to fit the horizontal position of the putter during the execution using the Darwinian Particle Swarm Optimization (*DPSO*) method.

In this paper, the parameters of the kinematic model (*i.e.*, amplitudes, frequencies and phases of the sinusoids) of each expert player's putting trial are classified with 5 different classification methods in order to identify possible links between different executions of the same player, thus extracting putting signatures for every player. The several classification algorithms studied are described in section 2, leading to the experimental results of the mentioned stages, that are represented and discussed in section 3. Finally, the main conclusions are outlined in section 4.

II. SURVEY OF CLASSIFICATION METHODS

In this section it is presented a review of the five classification algorithms used in this work. In particular, the Linear and Quadratic Discriminant Analysis (*LDA* and *QDA*), the Naïve Bayes with Normal (Gaussian) distribution and with Kernel Smoothing Density Estimate (*NV* and *NVK*) and the Least Squares Support Vector Machines with Radial Basis Function Kernel (*LS-SVM*).

Despite the diversity of the methodologies to evaluate the performance of classifiers, the confusion matrix and the area under the Receiver Operating Characteristic curve (*AUC*) are some of the most well-known methods to evaluate them and will be used in this work. Further details about both methods can be found in [13]. A confusion matrix is a matrix containing information about actual and predicted classifications done by a classification system [14]. The confusion matrix lists errors and successes in the test set. The main diagonal represents the correctly classified samples while the other elements of the matrix correspond to samples that were incorrectly classified. The Receiver Operating Characteristic (*ROC*) is a technique to

visualize, evaluate, organize and select classifiers based on their performance (Fig. 1). The *ROC* graphs can show the line between the true positive and false positive rate of the classifiers [15].

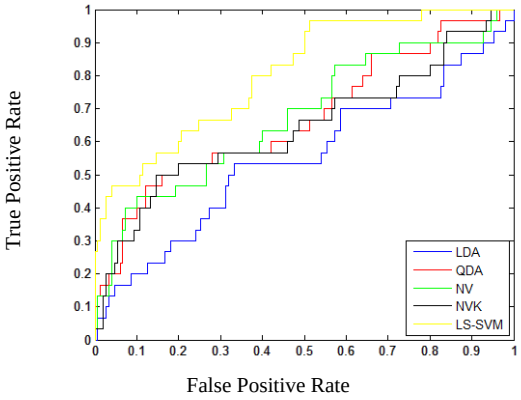

Figure 1. Example of *ROC* curves for the third sine wave (Amplitude (a_3) vs. Angular Frequency (b_3)) of the fifth player (class 5). The best possible prediction method would yield a point in the upper left corner, *i.e.*, coordinate (0,1), of the *ROC* space, hence the *LS-SVM* is the classifier which presents a superior performance for this trial.

To compare classifiers it is necessary to reduce the curve to a scalar value. A common method to achieve this reduction is to calculate the area under the *ROC* curve (*AUC*). The *AUC* is a way to measure classifiers performance. Since the *AUC* is a part of the area of the unit square, its value always varies between 0 and 1. An *AUC* value of 1 represents a perfect test while the *AUC* value of 0.5 represents a weak or worthless test.

A. *Linear Discriminant Analysis (LDA)*

There are many possible techniques for data classification. The Linear Discriminant Analysis (*LDA*) is one of the most commonly used techniques for data classification and dimensionality reduction in statistics, pattern recognition and machine learning, since it easily handles the case where the within-class frequencies are unequal and their performances has been examined on randomly generated test data [16]. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability (Fig. 2).

LDA is closely related to logistic regression, Principal Component Analysis (*PCA*) and Quadratic Component Analysis (*QDA*), which also attempt to express one dependent variable as a combination of other features or measurements. *LDA* looks for linear combinations of variables which best explain the data, by explicitly attempting to model the difference between the classes of data. Other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the *LDA* method. Hence, a difference between independent variables and dependent variables must be made. *LDA* assumes that the conditional probability density functions (*PDF*) follow a normal distribution for all classes. In practice, the class means and covariances are usually not known and are estimated from the training set used; *e.g.*, using the maximum likelihood estimate or the maximum a posteriori estimate. Also, *LDA* assumes that all classes have a common covariance matrix and all covariances have full rank, this is called the homoscedastic assumption. The discriminant function is given by testing the probability that a sample x is contained in one class or another:

$$\delta_k(x) = x^T \Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log P\,(C = k) \qquad (5)$$

Where, Σ is the covariance matrix common to all classes and μ_k is the mean of class k. The value of x in each $\delta_k(x)$ is calculated and the classification of x is the class k that yields the largest value. When the number of observations of each sample exceeds the number of samples, the covariance estimates do not have full rank, and so cannot be inverted. One way to deal with this is to use a pseudo inverse instead of the usual inverse matrix or use a Shrinkage estimator of the covariance matrix. The resulting classifier implies that the decision boundary between pairs of classes is linear and a hyperplane when using more than 2 classes, this is verified through the comparison of classes using the log-ratio. In geometrical terms, it is clear that an input observation is in a given class if the multidimensional-space observation point is located on a certain side of a hyperplane, perpendicular to the normal to the discriminant hyperplane.

LDA finds other applications in areas like face recognition, marketing or financial prediction. For more details on the implementation of the method, one should refer to [17].

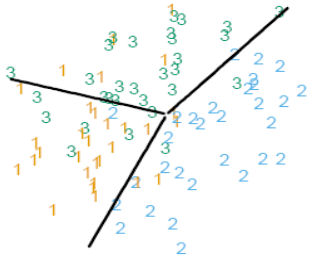

Figure 2. Example of fitted *LDA* decision boundaries separating three classes [17].

B. *Quadratic Discriminant Analysis (QDA)*

Although they differ in their derivation, Quadratic Discriminant Analysis (*QDA*) is similar to *LDA* [18]. The essential difference between them is in the way the linear function is fit to the training data (Fig. 3). Also very popular, *QDA* separates measurements of classes of objects or events with a boundary between each pair of classes described by a quadratic equation. Normal distributions are assumed, but each class can have a different covariance matrix. Thus, the separate covariance matrices must be estimated for each class. Which means there are much more parameters than in *LDA*, that increase with the number of dimensions. Since the decision boundaries are functions of the parameters of the densities, counting the number of parameters must be carefully done. When the homoscedastic

assumption is true, the best possible test for the hypothesis that a given measurement is from a given class is the likelihood ratio test, similarly to the *LDA*. The discriminant function is given by testing the probability that a sample *x* is contained in one class or another:

$$\delta_k(x) = -\frac{1}{2} log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log P\,(C=k) \qquad (1)$$

where Σ_k is the covariance matrix for class *k*, and μ_k is the mean of class *k*. After calculating each covariance matrix, estimating the mean, μ_k, and calculating $P(C = k)$, the classification of new samples is accomplished by calculating their discriminant function value for each class. The less rigid model underlying *QDA* may better approximate the situation in a real classification problem (compared to *LDA*). While the allowance of curved decision boundaries can lead to reduced bias in the estimation of the (unknown) optimal decision boundaries, having to estimate variance and covariance values with less data can lead to increased variance in the estimation of the optimal boundaries.

Literature show that both *LDA* and *QDA* perform well on large and diverse set of classification tasks.

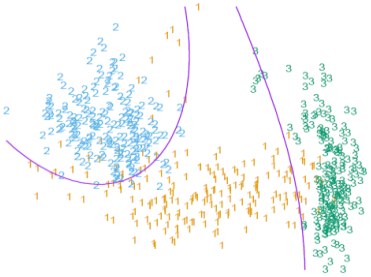

Figure 3. Example of fitted *QDA* decision boundaries separating three classes [19]

C. Naïve Bayes with Normal (Gaussian) distribution (NV)

Naïve Bayes (*NV*) is one of the most efficient and effective inductive learning algorithms for machine learning and data mining [19]. The *NV* classifier is designed for use when features are independent of one another within each class, which is a rather unrealistic assumption that is almost always violated in real-world applications. However, it has surprisingly good performance, performing well in practice even when that independence assumption is not valid. Furthermore, this assumption dramatically simplifies the estimation. The individual class-conditional marginal densities can be estimated separately; also if the variables are discrete, then an appropriate histogram estimate can be used. Assuming conditional independence among X_i's (vectors of observed random variables), Bayes Rules is given by:

$$P(Y = y_k|X_1 \dots X_n) = \frac{P(Y=y_k)\prod_i P(X_i|Y=y_k)}{\sum_j P(Y=y_j)\prod_i P(X_i|Y=y_j)} \qquad (2)$$

Hence, the Naïve Bayes classifier selects the class Y^{new} with maximum discriminant function for X_{new}=<X_1,…,X_n>:

$$Y^{new} \leftarrow \arg\max_{y_k} P(Y=y_k)\prod_i P(X_i^{new}|Y=y_k) \qquad (3)$$

If we have a continuous X_i, a common approach is to assume that $P(X_i \mid Y=y_k)$ follows a normal (Gaussian) distribution:

$$P(X_i = x|Y=y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_{ik}}{\sigma_{ik}}\right)^2} \qquad (4)$$

And the classification becomes:

$$Y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \mathcal{N}(X_i^{new}; \mu_{ik}, \sigma_{ik}) \qquad (5)$$

Where Π_k is estimated for each value of y_k by $\pi_k \equiv P(Y=y_k)$. Also, for each attribute X_i, it is necessary to estimate the class conditional mean μ_{ik} and variance σ_{ik}.

A well-known limitation of Naïve Bayes is in the case of binary features [20], where it can only learn linear discriminant functions, and thus it is always suboptimal for non-linearly separable concepts. Nonetheless, Naïve Bayes has proven effective in many practical applications, including text classification, medical diagnosis, and systems performance management [21]. Moreover, it is efficient using memory space and in terms of time complexity [22].

D. Naïve Bayes with Kernel Smoothing Density Estimate (NVK)

Kernel smoothing density estimation is an unsupervised learning procedure, which historically precedes kernel regression. This method fits a different but simple model separately at each query point x_0, using only observations close to the target point, in a way that the resulting estimated function becomes smooth. This is accomplished via a weighting function or kernel $K_\lambda(x_0;x_i)$ that assigns a weight to x_i based on its distance from x_0 [17]. Kernel methods use weights that decrease smoothly to zero with distance from the target point. In high-dimensional spaces the distance kernels are modified to emphasize some variable more than others.

The Kernel estimate is probably the most widely used nonparametric density estimation method. Kernels K_λ are typically indexed by a parameter λ that controls the width of the neighborhood. This results in a memory-based method that requires little or no training at all where all the work gets done at evaluation time. The only parameter that needs to be determined from the training data is λ. The model is the entire training data set. Generally the kernel function is given by:

$$K_\lambda(x_0, x) = PDF\left(\frac{|x-x_0|}{h_\lambda(x_0)}\right) \qquad (6)$$

Three popular probability density functions (*PDF*) used are Epanechnikov, Tri-Cube and Gaussian distribution function. A comparison of these kernels for local smoothing is presented in Fig. 4.

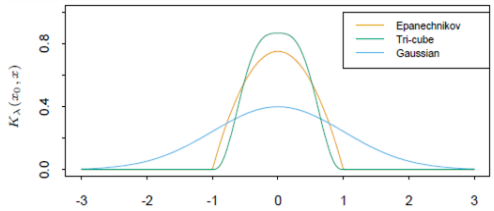

Figure 4. A comparison of three popular kernels for local smoothing [22].

Bayesian classification and decision making are based on probabilities that a given set of measurement come from objects belonging to a certain class (probability theory) and statistical methods based on class conditional probability density functions of features, are suitable in diverse classification tasks [23].

Estimated *PDF*s have been used for classification utilizing Bayes formula. The classification can be done based on the probability density function, instead of estimating posterior probability using Naïve Bayes. The attempt is to estimate the underlying density function from the training data, and the idea is that the more data in a region, the larger is the density function. Kernel smoothing density estimation leads naturally to a simple family of procedures for nonparametric density estimates for classification in a straightforward fashion using Bayes' theorem [17].

E. Least Squares Support Vector Machines with Radial Basis Function Kernel (LS-SVM)

Support Vector Machines (*SVM*) is a powerful methodology for solving problems in nonlinear classification, function estimation and density estimation which has also led to many developments in kernel based methods in general [24-27]. This method solves convex optimization problems, typically by quadratic programming. The Least Squares Support Vector Machines (*LS-SVM*) is a reformulation to the standard *SVM*s [28] which was recently proposed. In fact, when the data points are linearly independent, *LS-SVM* is equivalent to Hard Marginal *SVM* [29]. *LS-SVM* involves the equality constraints only. Hence, the solution is obtained by solving a system of linear equations.

SVM models are similar to multilayer perceptron neural networks. However, using a kernel function, *SVM*s are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training. Furthermore, rather than fitting nonlinear curves to the data, *SVM* handles this by using the kernel function to map the data into a different space where a hyperplane can be used to do the separation.

Many kernel mapping functions can be used but only a few have been found to work well in for a wide variety of applications. The default and recommended kernel function is the Radial Basis Function (*RBF*). According to [20], Kernel methods achieve flexibility by fitting simple models in a region local to the target point x_0. Localization is achieved via a weighting kernel K_λ, and individual observations receive weights $K_\lambda(x_0; x_i)$. Radial basis functions combine these ideas, by treating the kernel functions $K_\lambda(\mu; x)$ as base functions, where each basis element is indexed by a location and a scale parameter (μ_m and λ_m respectively). Thus, Radial basis functions are symmetric p-dimensional kernels located at particular centroids:

$$f_\theta(x) = \sum_{m=1}^{M} K_{\lambda_m}(\mu_m, x)\theta_m \qquad (7)$$

The centroids μ_m and scales λ_m have to be determined. A usual choice for the probability density functions is the standard Gaussian density function. There are also several approaches for learning the parameters μ_m, λ_m and θ_m. For example, a popular method is estimating θ_m, given μ_m and λ_m by a simple least squares problem. Often the kernel parameters μ_m and λ_m are chosen in an unsupervised way using the X distribution alone. One of the methods is to fit a Gaussian mixture density model to the training x_i, which provides both the centers μ_m and the scales λ_m. Figure 9 shows an example of Gaussian Radial Basis Function Kernels with scale parameter λ=1 and centered at 5 centroids, which were chosen at random.

The *LS-SVM* classifier was implemented using the *Least Squares – Support Vector Machines MatLab Toolbox* [30].

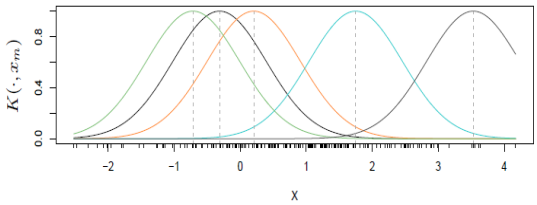

Figure 5. An example of a Gaussian Radial Basis Function Kernels centered at 5 centroids chosen at random and scale parameter λ=1[19].

III. Experimental Results

Experimental results present a comparative case study on the classification accuracy between five methods for the detection of signatures in the performance of the golf putting. In this stage, intensive *Matlab* simulation was performed using the detection algorithm and the *DPSO* as an estimation technique with the earlier defined parameters to obtain the putter's motion function that describes 30 putt executions of 6 different expert subjects (classes), in a total of 180 trials. After calculating all the estimation parameters, the five classification methods previously described were used in order to identify the signature of each player.

Figures 6-8 and Table 1 depict the confusion matrix and the area under the *ROC* curve (*AUC*) of the five classifiers, respectively.

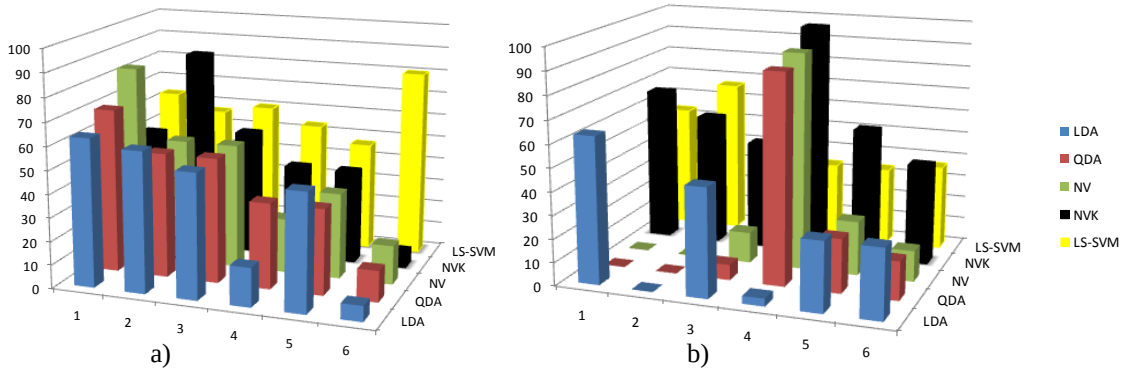

Figure 6. Comparisson of the percentual *TP* rate of the classifiers for the first sine wave.
a) Amplitude (a_1) vs. Angular Frequency (b_1); b) Amplitude (a_1) vs. Phase (c_1).

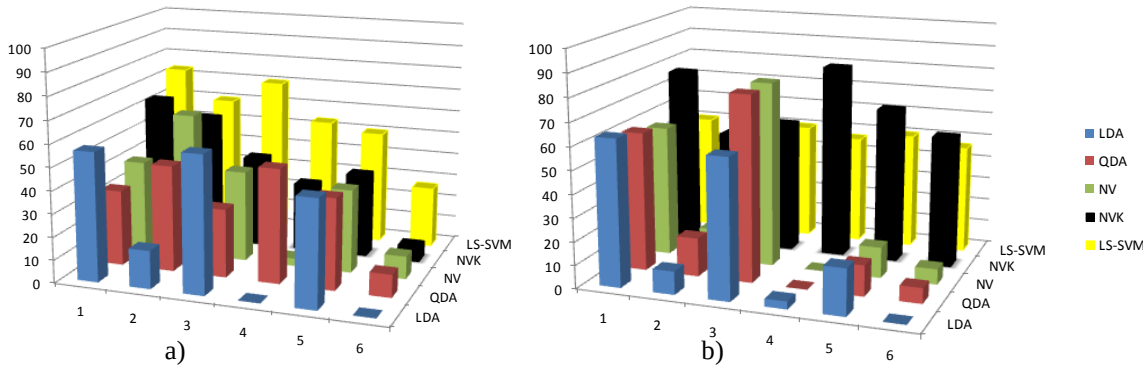

Figure 7. Comparisson of the percentual *TP* rate of the classifiers for the second sine wave.
a) Amplitude (a_2) vs. Angular Frequency (b_2); b) Amplitude (a_2) vs. Phase (c_2).

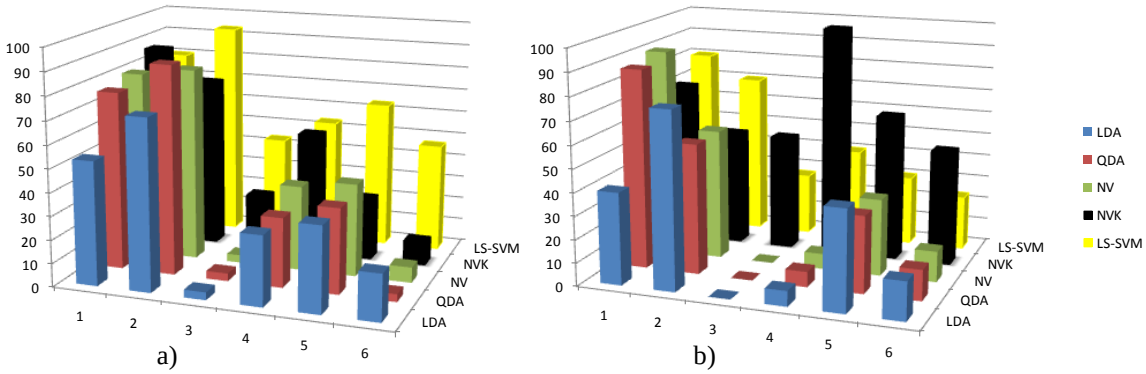

Figure 8. Comparisson of the percentual TP rate of the classifiers for the third sine wave.
a) Amplitude (a_3) vs. Angular Frequency (b_3); b) Amplitude (a_3) vs. Phase (c_3).

Based on the previous figures, the *LS-SVM* shows a better classification accuracy since it presents a higher percentage of true-positives (*TP*) in almost all situations closely followed by the *NVK* method.

However, in order to confirm the superiority of the *LS-SVM* over the *NVK*, the *AUC* of each player's trials was determined for the five classification methods. In order to allow a straightforward comparison of the five classifiers, next table depicts the average value of the *AUC* highlighting the maximum value for each player's trial. Once again, the previous tables and figures provide evidences about the superiority of the *LS-SVM* classifier which shows better results in the majority of the trials closely followed by the *NVK* classifier.

TABLE I. AVERAGE VALUE OF THE *AUC*.

Class	LDA	QDA	NV	NVK	SVM
1	0,619	0,601	0,671	0,680	0,744
2	0,650	0,623	0,692	0,685	0,737
3	0,566	0,582	0,634	0,761	0,734
4	0,507	0,585	0,574	0,675	0,690
5	0,622	0,651	0,692	0,766	0,797
6	0,493	0,602	0,650	0,718	0,745

IV. CONCLUSION

Real-time automated analysis of sport games such as football, tennis or golf is a domain receiving increased attention. Nowadays, in many live broadcasts, computer vision analysis, with special attention to the ball's kinematic, is used for example to present the ball's velocity or checking the ball's relative position. Also, relevance is given to information computed offline, like player's statistics.

The presented system for data retrieval, despite its complexity, is functional and allows retrieving a series of different information simultaneously. To validate the work, towards using it in real situations, the putting performance of six expert golf players was evaluated. In this work, several classification methods for golf putting performance analysis were used and compared. It was shown that *LS-SVM* showed the most consistent results. For this reason, in the future we intend to apply *LS-SVM* to extract unique features related to each player when performing the putt, in order to obtain a putting signature for every single subject.

REFERENCES

[1] Pelz, D. (2000). "Putting Bible: The complete guide to mastering the green". Publication Doubleday. New York.

[2] Alexander, D.L. and Kern, W. (2005). "Drive for show and putt for dough?" Journal of Sports Economics, 6, 1, 46-60.

[3] Jonassen, D.H. and Grabowski, B.L. (1993). "Handbook of individual differences, learning and instruction". NJ. Hillsdale: Lawrence Erlbaum.

[4] Delay, D., Nougier, V., Orliaguet, J.P. and Coello, Y. (1997). "Movement control in golf putting." Human Movement Science, 16, 5, 597-619.

[5] Coello, Y., Delay, D., Nougier, V. and Orliaguet, J.P. (2000). "Temporal control of impact movement: The "time from departure" control hypothesis in golf putting". International Journal of Sport Psychology, 31, 1, 24-46.

[6] Hume, P.A., Keogh, J. and Reid, D. (2005). "The role of biomechanics in maximising distance and accuracy of golf shots". Sports Medicine 35 (5): 429-49.

[7] Karlsen, J., Smith, G. and Nilsson, J. (2008). "The stroke has only a minor influence on direction consistency in golf putting among elite players." Journal of Sports Sciences, 26, 3, 243-250.

[8] Araújo, D. (2006). "Tomada de decisão no desporto." Lisboa: Edições FMH.

[9] Davids, K., Button, C. and Bennett, S. (2008). "Dynamics of skill acquisition – a constraints-led approach." Champaign. Illinois: Human Kinetics Publishers.

[10] Schöllhorn, W., Mayer-Kress, G., Newell, K.M and Michelbrink, M. (2008). "Time scales of adaptive behavior and motor learning in the presence of stochastic perturbations." Human Movement Science, 28, 3, 319-333.

[11] Phillips, E., Davids, K., Renshaw, I. and Portus, M. (2010). "Expert performance in sport and the dynamics of talent development." Sports Medicine, 40, 4, 271-283.

[12] Couceiro, M.S., Dias, G., Luz, J.M.A., Figueiredo, C.M., Ferreira, N.M.F. (2010). "Parameter Estimation for a Mathematical Model of the Golf Putting". WACI'10 - 5th Workshop on Applications of Computational Intelligence.

[13] Chan, K. and Lee, T.W. (2002). "Comparison of Machine Learning and Traditional Classifiers in Glaucoma Diagnosis". IEEE Transactions on Biomedical Engineering, Vol. 49, Nº 9, September.

[14] Kaladhar, D.S.V.G.K., Nageswara Rao, P.V. and Ramesh Naidu Rajana, B.L.V. (2010). "Confusion matrix analysis for evaluation of speech on Parkinson disease using Weka and MatLab", International Journal of Engineering Science and Technology, 2(7):2733-2736.

[15] Fawcett, T. (2006). "An introduction to ROC analysis". Pattern Recognition Letters, Vol. 27, Issue 8, June.

[16] Martínez, Aleix M. and Kak, Avinash C. (2001). "PCA versus LDA". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, Nº 2, February.

[17] Hastie, T., Tibshirani, R. and Friedman, J. (2009). "The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction". Springer Series in Statistics Springer: 2nd ed.

[18] Roushanzamir, S., Valafar, H. and Valafar, F. (1999). "A comparative study of linear and quadratic discriminant classifier techniques for variable selection: a case study in predicting the effectiveness of hydroxyurea treatment of sickle cell anemia". IJCNN '99 - International Joint Conference on Neural Networks.

[19] Zhang, H. (2005). "Exploring conditions for the optimality of naive Bayes". International Journal of Pattern Recognition and Artificial Intelligence, Vol. 19, No. 2.

[20] Rish, I. (2001). "An empirical study of the naive Bayes classifier". IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.

[21] Domingos, P. and Pazzani, M. (1997). "On the optimality of the simple Bayesian classifier under zero-one loss". Machine Learning, 29:103–130.

[22] Zhang, H., Ling, C. X. and Zhao, Z. (2000). "The Learnability of naive Bayes", in Hamilton H. and Yang Q.(eds), Advances in Artificial Intelligence (Proceedings of CAI2000), Vol. 1822/2000, 432-441, Springer.

[23] Rahimizadeh, H., Marhaban, M.H., Kamil, R.M., and Ismail, N.B. (2009), "Color Image Segmentation Based on Bayesian Theorem and Kernel Density Estimation", European Journal of Scientific Research, EuroJournals Publishing, Inc. 2009, Vol.26 No.3, pp.430-436.

[24] Vapnik, V. (1998). "Statistical learning theory", John Wiley, New-York.

[25] Scholkopf, B., Burges, C. and Smola, A. (1998). "Advances in Kernel Methods - Support Vector Learning", MIT Press.

[26] Cristianini, N., and Shawe-Taylor, J. (2000). "An Introduction to Support Vector Machines", Cambridge University Press.

[27] Scholkopf, B. and Smola, A. (2002). "Learning with Kernels", MIT Press, Cambridge MA.

[28] Suykens, J.A.K. and Vandewalle, J. (1999). "Least squares support vector machine classifiers", Neural Processing Letters, 9(3), 293-300.

[29] Ye, J. and Xiong, T. (2007). "SVM versus Least Squares SVM". In Proc. Of The Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007), pp. 640–647, March 21-24, San Juan, Puerto Rico, U.S.A.

[30] Pelckmans, K., Suykens J. A. K., Gestel, T. V., Brabantar, J. D., Lukas, L., Hamers, B. and Vandewalle, J. (2002). "LS-SVMlab: A MATLAB/C Toolbox for Least Squares Support Vector Machines". Leuven-Heverlee, Belgium.