A Gaussian Groundplan Projection Area Model for Evolving Probabilistic Classifiers

Theodoros Theodoridis
School of CSEE
University of Essex
Wivenhoe Park, Colchester
CO4 3SQ, U.K.
ttheod@essex.ac.uk

Alexandros Agapitos
Natural Computing Research and Applications Group
University College Dublin, Ireland
alexandros.agapitos@ucd.ie

Huosheng Hu
School of CSEE
University of Essex
Wivenhoe Park, Colchester
CO4 3SQ, U.K.
ttheod@essex.ac.uk

ABSTRACT

In this paper, an investigation of evolvable probabilistic classifiers is conducted, along with a thorough comparison between a classical Gaussian distance model, and the induction of Gaussian-to-circle projection model. The newly introduced model refers to a distance fitness measure, based on the projection of Gaussian distributions with geometric circles. The projection architecture aims to model and classify physical aggressive behaviours, by using biomechanical primitives. The primitives are being used to model the dynamics of the aggressive activities, by evolving biomechanical classifiers, which can discriminate between three behaviours and six actions. Both evolutionary models have shown strong discrimination performances on recognising the individual actions of each behaviour. From the comparison, the proposed model outperformed the classical one with three ensemble programs.

Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: Automatic Programming

General Terms

Algorithms, Performance, Experimentation

Keywords

Action Recognition, Gaussian Fitness Model, Biomechanical Primitives, Time Series Classification

1. INTRODUCTION

For the multiclass problem, the goal is to evolve expression programs that suggest single numeric values denoting the class pattern being recognised. The patterns we investigate regard aggressive physical activities described by the following behaviours: Upper (armStrikes) = {punching[1], slapping[2]}, Inertial (bodyStrikes) = {pushing[3], pulling[4]}, and Lower (legStrikes) = {front-kicking[5], side-kicking[6]} (see Fig. 1). Using a set of biomechanical primitives, depicted in Table 1, GP is designated to construct a solution capable of discriminating among these patterns.

Our approach presents a novel methodology, extension of the work of [1, 2, 3] (who have used the GM model), where patterns of Gaussian distribution are treated as geometric circles over a ground-plan view. The methodology exploits the geometric properties of a circle, which virtually represents a distribution. The process estimates the overlap of the intersected areas for every two classes. Such distributions are being used to model the behaviour of each program, based on the training examples for each class. A program output distribution can be modeled as a mixture of normal distributions [1]. Clearly, a good program will produce distant output distributions for examples of different classes. A model of each program output distribution, for a particular class, can be acquired by evaluating the program on the example training set. This is accomplished by taking the mean and the standard deviation of the program outputs for those training examples. The ultimate goal of the system's architecture is to translate the numerical output of the GP classifier into a class label.

Previously, in [2] the binary problem is decomposed in two binary subproblems, thus letting GP to evolve a solution based on the class being tested. Two dynamic boundary determination approaches based on centre and slotted boundaries, suggested by [4], utilise single numeric output, and a class label is given after some transformation. Moreover, [5] suggested how evolutionary fusion methods using GP can perform well when combining classifiers of different nature. They suggest that their methodology differs significantly from any other ensemble method [6]. The Gaussian model (GM), coined by [1], uses a probability based approach of Gaussian distributions. The method constructs fitness evaluations that assess the classification performance with the distribution distance, and the overlapping area. This twofold behaviour models each class with a Gaussian distribution, using a number of training cases. The distance/area fitness model generates Gaussian probabilities, which provide recognition estimations for each class.

Paper outline: Section 2 presents analytically the Gaussian ground-plan projection area model. In section 3, the evolutionary algorithm, variation operators, run parameters, and the fitness function are given. Experimental results are demonstrated in section 4, and conclusions in section 5.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

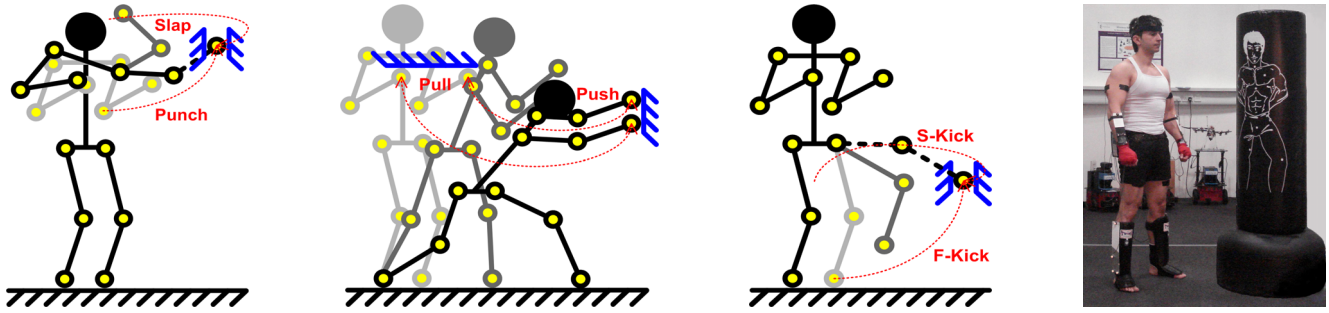


Figure 1: Free body diagrams of the three biomechanical aggressive behaviours. (a) Upper: arm-strikes (punching, slapping), (b) Inertial: body-strikes (pushing, pulling), (c) Lower: leg-strikes (front-kicking, side-kicking), (d) Sensor placement and standing bag.

2. GGPA FITNESS MODEL

The Gaussian Groundplan Projection Area model (GGPA), accounts Gaussian distributions as circles on a ground plan (imaginary) view. The distributions are being evaluated from the top as depicted in Fig. 2(a). Such representation makes use of geometric circles, by associating the dimensional properties of a Gaussian as the mean μ and the standard deviation σ, with the centre c and the radius r of a circle respectively.

Based on this concept, during the evolutionary process each individual is evaluated through a number of training cases for each class. Hence, every pattern forms a normal distribution of evaluations, acquired by the distance metric of Eq. 6, with samples equal to the number of training cases. From such Gaussian distributions, only the first σ, which corresponds to $\mu \pm 1\sigma \equiv 68\%$, is exploited as it is the area with the most essential samples that includes the most significant evaluations. The model, for each distribution, surrogates a geometric circle to represent its 68% of samples, in a ground plan projection (see Fig. 2(a)). A congruence relation is established between the centre of the circle c and its radius r, equaling with the distribution's μ and σ respectively. The relation is described as follows: $\mu \equiv c$, $\sigma \equiv r$.

A repulsive weight $w = 1/d$ is employed to equally push the two Gaussians away from each other; thus, the closer the distribution means are, the more repulsive the weight becomes. This is analogous to the distance d (see Eq. 1), pushing the distributions to opposite directions as depicted by Fig. 2(b).

$$d = |\mu_1 - \mu_2| \tag{1}$$

$$x_1 = \frac{d^2 + r_1^2 - r_2^2}{2d} \tag{2}$$

$$x_2 = \frac{d^2 + r_2^2 - r_1^2}{2d} \tag{3}$$

$$a(x, r) = \frac{1}{2}\pi r^2 - x\sqrt{|r^2 - x^2|} - r^2 \cdot \arcsin\left(\frac{x}{r}\right) \tag{4}$$

Eqs. 2 and 3, represent the circle's geometry of the distance between the centre c and the edge of the half plane. The area of intersection is given by Eq. 4, which computes the overlap covered by the two circles. The overlap or intersection area is also defined by the shared chord o, presented by Fig. 2(b), with the two edge points coming from the two circles denoting the intersection instances [7].

Apart from the circle intersection, which can occur when two pattern Gaussians are relatively close, there is a number of extreme contingencies (see Fig. 2(c)). In the first case, the Gaussians have almost equal means so their circles are under complete overlap. In the second, worst case scenario, the two Gaussians ideally do not overlap at all so there is no intersection between the circles. This means that there is a significant separation of the two compared patterns. Fig. 2(c) illustrates this notion with three different Gaussians differing in σ, with Gaussian μ_1 being the pattern distribution to be compared with μ_2 and μ_3. Thence, the Gaussians $\mu_{1,2}$ present absolute overlap as observed by circle c_1 embedded in c_2; this contingency is described by Eq. 5(b), where in this case the inner circle area is returned. On the other extreme, Gaussians $\mu_{1,3}$ appear no overlap, and according to Eq. 5(a), the area of intersection returns zero. In the case of normal overlap, the sum of the two intersected areas is returned (see Eq. 5(c)).

$$area(d) = \begin{cases} 0 & \text{if } d \geq r_1 + r_2 \quad (a) \\ \begin{cases} \pi r_1^2 & \text{if } r_1 < r_2 \\ \pi r_2^2 & \text{otherwise} \end{cases} & \text{if } d \leq |r_1 - r_2| \quad (b) \\ a(x_1, r_1) + a(x_2, r_2) & \text{otherwise} \quad (c) \end{cases} \tag{5}$$

$$A(d) = \frac{1}{\frac{1}{d} + area(d)} \tag{6}$$

For the sake of convenience, the area derived by Eq. 5 is normalised by the fraction of Eq. 6. This is a distance-based metric, as well as the raw fitness function of the evolutionary algorithm. According to this normalisation, overlapping distributions are evaluated with even close to zero distance. In practice, the further their means are, the more the distance increases exponentially[1], as long as there is overlap. For distributions that do not overlap at all, the normalised distance increases linearly as portrayed by Fig. 2(d). From this figure, the distribution in black (range $\{0,\ 10\}$), represents the pattern to be compared with the inner distribution in gray (range $\{2, 8\}$). The light gray distributions de-

[1]The exponential growth of the separated means is a behaviour that emerged from the model's design. We declare that we did not fit an exponential curve to the rate of growth to produce this behaviour. It was a purely random effect.

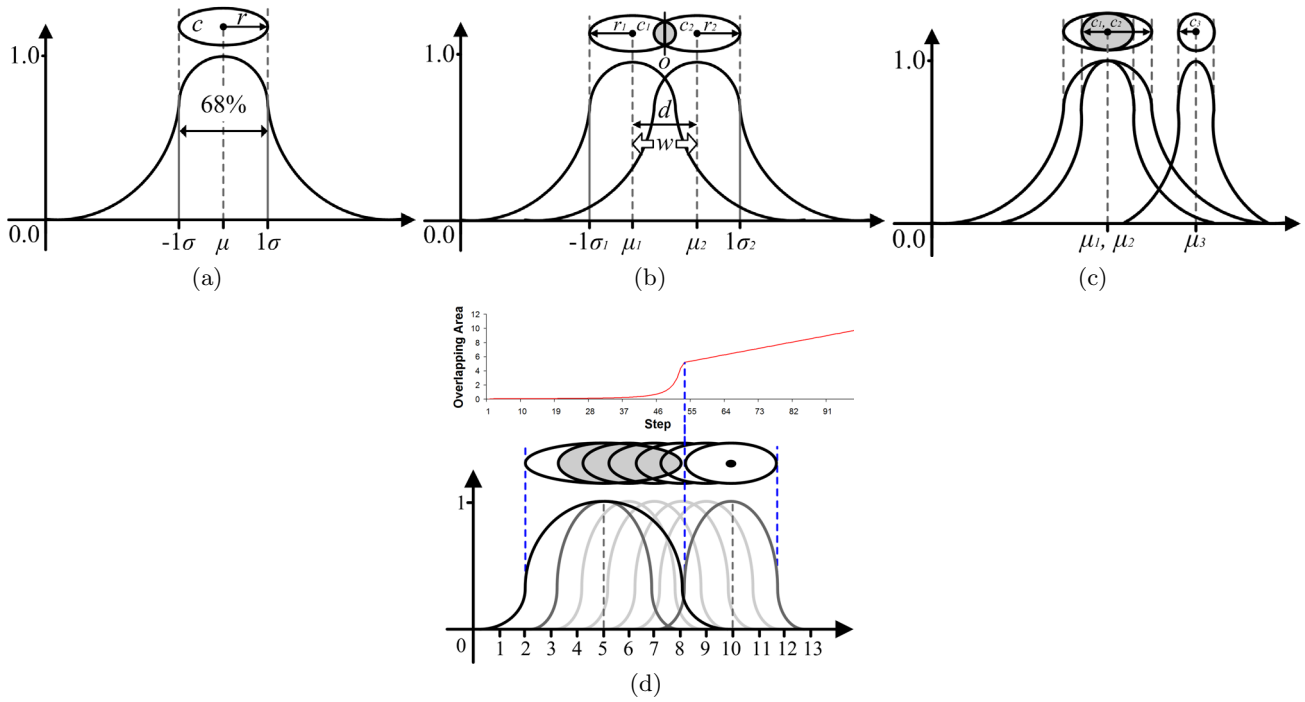


Figure 2: (a) Ground plan projection, (b) Gaussian intersection, (c) Extremes, (d) Overlapping areas.

note a displacement of the inner one towards the right side, which ultimately ends with zero overlap. To test this sliding behaviour, 10 steps have been performed with fraction 0.1 equaling to 100 stepwise displacements to the right side. The exponential growth of the distance (steps 1 to 54), shown in the figure, appears when overlap occurs. Further, the distance of the overlapping area increases linearly (steps 55 to 100), when there is no intersection. Such twofold behaviour of the normalised area $A(d)$ is particularly useful since it can perform a detailed separation when overlap is involved between two distributions. Eventually, the distributions are crafted analytically until separation occurs, entailing with minimal circle overlap.

· ***Binomial Fitness Function of Gaussians***

Assuming a binary problem case, the fitness function is used to determine the *overlapping/intersection area* as a distance metric between classes i and j, similar to [1, 2], by correlating the μ with c and the σ with r. In multiclass pattern classification, the fitness function is determined by considering the distribution distance between every two classes. For the N-class problem, there are $C_N^2 = N!/2!(N-2)!$ class combinations, and the fitness function takes the form:

$$fitness = \frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{C_N^2}\frac{1}{1+A(d)_{ij}} \tag{7}$$

where T defines the training examples, N is the number of classes, $A(d)_{ij}$ is the distance of the intersection area for the index of the training examples i, and class combinations j[2].

[2]The indexes i and j correspond to identifiers 1 and 2 respectively as shown in Eqs. 1 through 5.

· ***Model Implementation***

Training Phase: At the initial stage, the GGPA fitness measure is applied on the GP system as follows: the individual to be evaluated uses its evolved inputs (≤ 27), where the training time series data are fetched into. Each input represents a biomechanical primitive from the set shown in Table 1. Thereafter, the individual is evaluated for T training cases, and from these evaluations a Gaussian normal distribution is formed for each class. Consequently, the μ and σ are estimated. The GGPA model then is applied on these distributions through Eq. 6. As mentioned earlier, the distributions properties are being regarded as geometric properties of circles ($\mu \equiv c, \sigma \equiv r$). If two class distributions overlap, then similarly the represented circles should intersect. The overall distance (error) is estimated by applying binomial search for every 2 Gaussians (see Eq. 7), and the average distance is returned. This distance constitutes the individual's fitness.

Testing Phase: At the final stage, the testing process of the individual's classification accuracy is conducted as follows: the probability density function of Eq. 10 is employed, to generate the class probabilities from the class distributions acquired from training earlier. The μ and σ are being applied on the Pdf, with the parameter x representing the individual's evaluation to a newly presented test instance. Thence, a number of evaluation probabilities will be produced, equal to the number of test cases for each class. The highest class probability determines the class being tested.

3. EVOLUTIONARY PARAMETERS

3.1 Program Representation Language

Table 1 presents the strongly-typed programming language, used for the construction of probabilistic classifiers. The

`List` argument type of the six biomechanical features, receives the whole time series signal from a parameter input. For each primitive, a slope sign change method based on the directional changes of a vector (ex: up-to-down, or left-to-right movement traversals), undertakes to break the signal in n fragments that is equal to the number of extracted slopes. Thereafter, a biomechanical evaluation of a primitive is estimated for every chunk, and the maximum value among all is returned in `double`. The arithmetics are being used to induce a variety of alterations of the primitive estimates, by using 50 constants from the set {0, 1} (step 0.02). Lastly, the conditional structures are employed to induce comparisons of the primitive values, that lie into representative intervals for each class.

Table 1: Primitive language for evolving probabilistic classifier programs.

Feature	Equation		Argument(s) type	Returns
(α) acceleration	$[u(t) - u_0(t)]/\Delta t$	(m/s^2)	List	double
(d) displacement	$u(t)\Delta t$	(m)	List	double
(F) force	$m\alpha(t)$	(N)	List	double
(J) impulse	$F\Delta t$	(Ns)	List	double
(P) power	$Fu(t)$	(W)	List	double
(T) energy	$1/2mu^2(t)$	(J)	List	double

Arithmetic	Argument(s) type	Returns
+, −, ∗, /	double, double	double

Conditional	Argument(s) type	Returns
If-Then-Else	boolean, double, double	double
and, or	boolean, boolean	boolean
>, <	double, double	boolean

Terminal	Value	Type
Constants	50 reals in range {0, 1} step 0.02	double
Parameters	27 time series	double

3.2 Evolutionary Algorithm

Our evolutionary algorithm (EA) is an elitist, generational model genetic algorithm, employed in accordance with a panmictic population mating scheme. The evolutionary run proceeds for 100 generations, and the population size is set to $1,000$ individuals. Evolution halts when all of 100 generations have elapsed. Random sampling of full expression-program trees has been set for the population initialisation, whereas for the evolutionary process grow trees are sampled. The use of only grow trees aims to the creation of less bushy expressions, which engage large amounts of memory and consequently delay the evolutionary runs. With initial depth of 5, the trees are allowed to grow up to the depth of 17. During fitness assignment, each program is being evaluated with 27 parameters[3] representing the input time series.

The EA employs a mutation-based variation scheme to explore the space with probability 0.9, whereas the rest 0.1 is set to perform Koza's sub-tree crossover [8]. A heuristic search scheme is defined by a probabilistically governed application, based on a mixture of standard mutation variation operators as initially presented by Chellapilla [9]. The λ[4] factor of the multi-mutation architecture has been set equal to 6. Unlike [9], the multi-mutation operators introduced by Eq. 8, contain three groups of operators which (a) enhance diversity: `grow` and `fair`, (b) control bloating: `hoist`, `permut`, and `fair`, and (c) perform smooth variability: `point` and `term`. Eventually, a tiny probability of 0.0001 has been set for reproduction, and when it is false the above search methods are executed.

$$Offspring = \texttt{hoist}(\texttt{permut}(\texttt{point}(\texttt{term}(\texttt{fair}(\texttt{grow}(Parent)))))) \quad (8)$$

For the selection mechanism, tournament selection has been used with a dynamically adaptive selection pressure to promote exploration for the early generations, and exploitation for later ones. Consequently, Eq. 9 uses the factor 0.1 to adjust the range of the individuals being engaged from the first to the last generation. With the parameters given above, the adaptive selection pressure $s_p(g)$ ranges in the interval {2, 100} individuals, incrementing up to 1% per generation. In addition, negative tournament is also employed by replacing the worst individual with the elitist.

$$s_p(g) = \begin{cases} 2 & \text{if } \lceil (0.1\frac{g}{G}) \cdot P \rceil < 2 \\ \lceil (0.1\frac{g}{G}) \cdot P \rceil & \text{otherwise} \end{cases} \quad (9)$$

where g is the current generation, G is the total generation number, and P is the population size.

Similar to [10], who proposed an alternative sampling behaviour based on tournament selection to perform significant computational savings, our EA utilises a boosting-resource-saving method to enhance the speed of the generations. This method targets on the fitness evaluations, by neglecting the individuals being evaluated more than once within the breeding loop of a generation. For the implementation of this notion, we assign a boolean flag to `true` to the individuals who have already been evaluated, so they cannot be re-evaluated if picked again by the selection mechanism.

3.3 Ensemble Pattern Classification

In [1, 2], they have used the *Joint Probability* (JP, Eq. 11) to create ensembles of *multiple best programs*, based on the probability density function (Eq. 10) that measures which class belongs to a given pattern. JP has been used to enhance the classification accuracy of a Gaussian model. Similarly, we have used a number of formal ensemble methods, along with JP, such as the *Distribution Summation* (DS, Eq. 12), and the *Majority Voting* (MV, Eq. 13). In addition, we introduce a new weighting method, the *Statistical Voting* (SV, Eq. 14), to be used as well with the GGPA model.

$$Pdf(\mu,\ \sigma,\ x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (10)$$

$$E_{JP}(x) = \underset{c_i\ \in\ C_N}{argmax} \prod_{m=1}^{M_N} P_m(Pdf = c_i|x) \quad (11)$$

$$E_{DS}(x) = \underset{c_i\ \in\ C}{argmax} \sum_{m=1}^{M_N} P_m(Pfd = c_i|x) \quad (12)$$

$$E_{MV}(x) = \underset{c_i\ \in\ C}{argmax} \left[\sum_{m=1}^{M_N} g(P_m(Pfd = c_i|x), c_i)\right] \quad (13)$$

$$E_{SV}(x) = \underset{c_i\ \in\ C_N}{argmax} \left[\ \frac{1}{C_N} \sum_{i=1}^{C_N} \left(\sum_{m=1}^{M_N} g(P_m(Pdf = c_i|x), c_i)\right)^p\ \right]^{1/p} \quad (14)$$

[3]The set of axis coordinates $\{x,\ y,\ z\}$ times 9 marker sensors coming from the head, elbows, wrists, knees, and ankles, equal to 27 input parameters.

[4]The λ factor denotes the number of random mutation operations being applied to an individual.

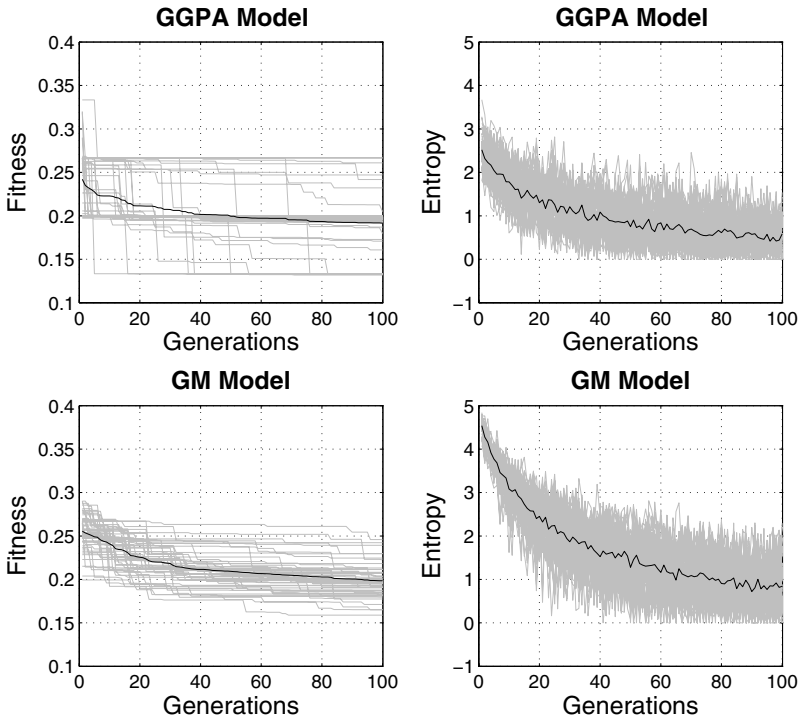


Figure 3: Fitness and entropy training plots for the GGPA and GM models.

where m is the classifier model, tested with M_N max number of models, x is the model's output, and P_m is the probability Pdf of class c_i (C_N = max number of classes), given a testing instance x [6]. Function $g = 1$ if $P_m = c$, and 0 otherwise.

The statistical voting uses the generalised mean for assessing distributions of votes, which have similar values. We set the generalised exponent relatively high ($p = 10$), so distributions with more votes to be rewarded. Eventually, the class with the highest probability, derived from the ensemble assessments, is designated as the class of the pattern. All the above ensemble methods induce 2, 3, and 4 combinations using binomial search with expansion depth of ten best programs.

4. EXPERIMENTAL RESULTS

· ***Protocol***

Three male and two female subjects (age 25 to 30), who have experienced aggression in scenarios such as physical fighting, took part in the experiment. Throughout six individual experiments, each subject had to perform the six aggressive activities mentioned in Section 1. Regarding the rights of the subjects involved, ethical regulations and safety precaution have been followed based on the code of ethics of the British psychological society [11]. The regulations explain the ethical legislations to be applied when experiments with human subjects are conducted. According to the experimental setup and the precautions taken, the ultimate risk of injuries was minimal. Following similar experimental procedures as in [12], the subjects were aware that their involvement in this series of experiments was voluntary, and it was made clear that they could withdraw at any time.

· ***Instrumentation***

The Essex robotic arena was the main experimental hall where the data collection took place. With area 4×5.5m, the subjects expressed aggressive physical activities at random locations. A professional kick-boxing standing bag has been used, 1.75m tall, with a human figure drawn on its body (see Fig. 1(d)). The subjects' performance has been recorded by the Vicon's nine ubiquitous cameras, interfacing human activity with spatial coordinate points. Based on this context, the data acquisition process involved nine reflectable markers placed on the forearms (elbows and wrists), four on the forelegs (knees and ankles), and one on the top of the head. Each group of markers constitutes a kinematic model.

· ***Data Setup***

From the overall number of markers, there are 27 input time series for all the three x, y, and z coordinates (9 markers $\times$ 3 coordinates). Each marker-coordinate time series contains $\sim 3,000$ samples ($\sim$ 15 actions per experimental session for each subject), which has been normalised in the interval $\{0,\ 1\}$. For training we used the 60%, first 1800 samples of the recorded time series, and for testing we reserved the remaining 40% (1200 samples). The training of each classifier was based on the physical performance of all the ten subjects, including a sample of about 90 physical actions, and for the testing, 60 actions were employed for each class in overall. To test the generalisation performance of the evolutionary classifiers, a 5-fold cross validation has been used. Thereupon, the training percentages from each fold-subject has been given to the GP system to evolve a classifier program, whereas the testing percentages assessed its generalisation performance.

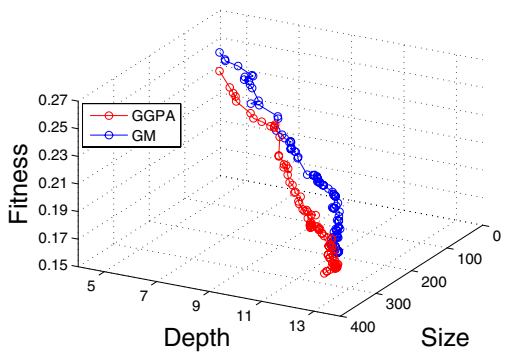
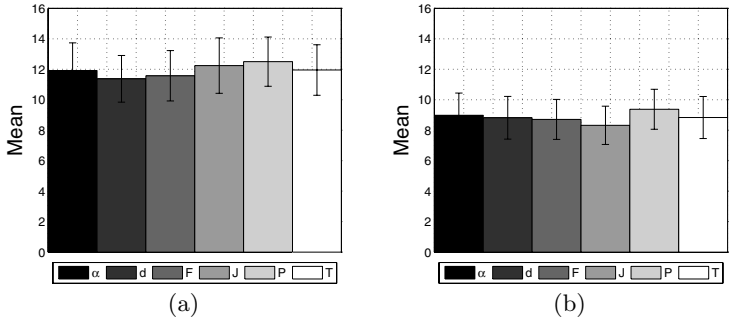


Figure 4: Bloating paths of the GGPA (red), and GM models (blue).



∗(d) Displacement, (F) Force, (J) Impulse, (P) Power, and (T) Energy.

Figure 5: Average error bars of the six evolvable biomechanical primitives, extracted from each probabilistic model via 50 simulations. (a) GGPA model, (b) GM model.

4.1 Training Performance

Comparing the training fitness graphs between the GGPA and the GM models from Fig. 3, we observe that the average training error is diminished up to 0.2, for both cases, with signs of stagnation after generation 40. However, despite the similar average training errors, the variation of the fitness graphs for all the 50 runs for the GM model, seems to converge to a more smooth direction. On the other hand, the phenotypic diversity measure depicted by the entropy graphs, demonstrates that the fitness entropy (Eq. 15) over the generations, showed a diminishing trend for both cases. This is an indication of decreasing diversity in the population of programs. Both models stabilise the production of diverse individuals with unique phenotypic-based solutions after generation 80.

$$-\sum_{k} p_k \cdot \log p_k \tag{15}$$

where p_k is the proportion of the population P, and k is the occupied partition [13].

Another worth looking measure of the training performance between the two models is to compare the three fundamental parameters that have been promoted by the evolutionary process. These are the *size*, *depth*, and *fitness* from which we can study whether the two models suffered from bloat. We caption the plot of the three parameters in a 3D graph as the *bloating paths*, shown by Fig. 4.

In the figure, the bloating paths show the average values estimated throughout 50 independent runs. Comparing the two paths, we see a very close resemblance of the linear direction they have followed. Remarkably, both models did not evolve trees with depth more than 12, and size around 300 primitives. For all the runs, the average fitness kept a decreasing behaviour, which complements to the models overall performance that they did not suffer from bloat. Owing to the considerable resemblance, we have performed a t-test between all the paired parameters to elicit which parameter differs significantly. The t-tests results pointed out that the sizes of both models are significantly different with $p = 1.04 \times 10^{-5}$ ($p < 0.05 \neq H_0$[5], $df = 198$, $sd = 104.27$). The depths also differ significantly with $p = 0.017$ ($p < 0.05 \neq H_0$, $df = 198$, $sd = 2.53$), as well as the fitnesses with $p = 2.70 \times 10^{-8}$ ($p < 0.05 \neq H_0$, $df = 198$, $sd = 0.013$).

From these results, we see that despite the linearity that prevented the generations to bloat, the models followed slightly different strategy to preserve the size and depth analogous to the fitness. In average, the training performance of the probabilistic classifier models pursued a very similar and smooth evolutionary progress, which does not always reflect though to analogous testing performances. The following section demonstrates the discriminant performance of the models, tested by unseen activity data.

The average biomechanical primitives that have been used over the 50 evolutionary runs, evolved from each classifier model, are demonstrated in Fig. 5. Each model has used a different number of primitives, by preserving a relatively equal amount of features for each category. Fig. 5(a) indicates that the *power* and *impulse* have dominated for the GGPA model, with average amount of primitives reaching the 12. Conversely, in Fig. 5(b) the average number of primitives for the GM model was 9, with the *power* energetic feature being distinguished as well. The commonality of this primitive appearing in both models, does not however differ significantly from the rest primitives. It is intriguing though that from this series of experiments, the evolutionary models used more this primitive. Evidently, this tells us that the subject's performances was characterised as physically intensive, and consequently violent. A final remark to the reason why the GGPA model engages more primitives than the GM is due to another random behaviour, that emerged stochastically.

4.2 Testing Performance

For the testing performance, we picked the elitist best-of-run individuals from each model (GGPA, GM). The elitist models have been evaluated with five performance measures that assessed the classification ability of the classifiers. Tables 2(a) and 2(b) depict the analytical and overall performance of the GGPA and GM models. Starting with Table 2(a), the overall performance indicates that the GGPA model made use of the SV ensemble method (Eq. 14) to evolve highly discriminant classifier programs. The ensemble combinations promoted 3 programs with indexes

[5] $\neq H_0$ denotes that the null hypothesis is rejected.

Table 2: Analytical and overall classification statistics. The parentheses in the *Program No.* denote the indexes of the best-of-run elitist individual(s), selected from a binomial search method from 50 independent runs. (a) GGPA model, (b) GM model.

(a)

Analytical Performance									Overall Performance	
Behaviour	Action	sub1	sub2	sub3	sub4	sub5	**AVG** %	AVG Behaviour %	Measure	Value
Lower Inertial Upper	Punch	1^1	1^1	1^2	×	1^1	$80^{0.33}$	$80^{0.33}$	*Precision*	1.000
	Slap	×	2^1	2^2	2^1	2^1	$80^{0.33}$		*Recall*	0.733
	Push	3^2	×	3^2	3^2	3^2	$80^{0.53}$	$80^{0.39}$	*Accuracy*	0.733
	Pull	4^1	4^1	4^1	×	4^1	$80^{0.26}$		*F-Measure*	0.846
	F-Kick	5^1	5^3	5^2	5^1	×	$80^{0.46}$	$90^{0.49}$	*G-Mean*	0.856
	S-Kick	6^3	6^1	6^1	6^1	6^2	$100^{0.53}$		Ensemble Method	SV
	AVG %	$83.3^{0.44}$	$83.3^{0.38}$	$100.0^{0.55}$	$66.6^{0.27}$	$83.3^{0.38}$			Program No.	3 (9,2,0)

(b)

Analytical Performance									Overall Performance	
Behaviour	Action	sub1	sub2	sub3	sub4	sub5	**AVG** %	AVG Behaviour %	Measure	Value
Lower Inertial Upper	Punch	×	1	1	×	1	50.0	25.0	*Precision*	1.000
	Slap	×	×	×	×	×	0.0		*Recall*	0.567
	Push	×	×	×	3	3	33.3	66.5	*Accuracy*	0.567
	Pull	4	4	4	4	4	100.0		*F-Measure*	0.708
	F-Kick	×	×	×	5	5	33.3	66.5	*G-Mean*	0.744
	S-Kick	6	6	6	6	6	100.0		Ensemble Method	None
	AVG %	33.3	50.0	50.0	66.6	83.3			Program No.	1 (14)

9, 2, and 0, out of 50 elitists (0-49). From Table 2(a), the combination of three programs appears to produce better classification accuracy, as also verified by [1]. The recognition *Accuracy* of the correctly classified instances reached the 73.3%, whereas the exactness of the classified instances (*Precision*) got 100%, for the reason that no out-of-scope misclassified instances were identified (scope of the action set: {1, 2, ..., 6}). Moreover, the completeness of the classification (*Recall*) got analogous score. Finally, the high proportionality estimations of *F-Measure* and *G-Mean*, profess an overally balanced classification performance.

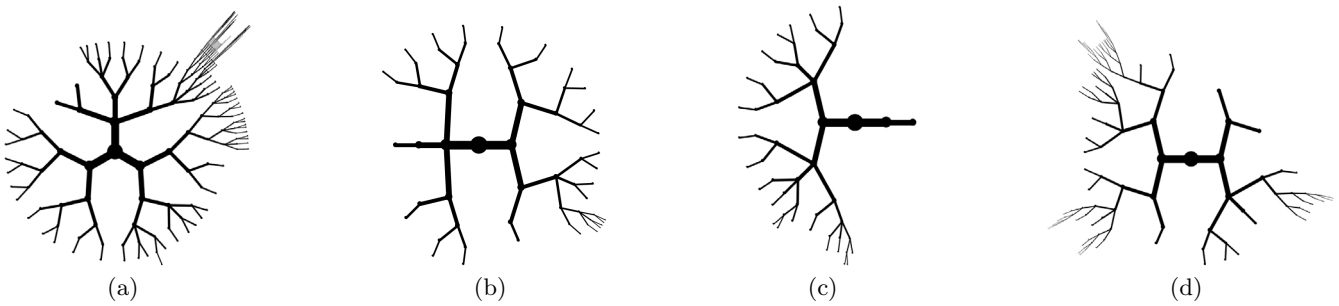
The analytical performance shows that the GGPA model managed to identify at least one or more actions from each subject, as indicated by the $ClassNo^x$, where $x \in \{1, 4\}$ ensemble programs. Conversely, the "×" symbol denotes that none of the models managed to correctly identify an action. The averages for each column are descriptive indications of the ensemble performance; what really the ensemble method derived is given by the overall performance. The average descriptive results have been estimated by Eq. 16.

$$y^z : \; y = \frac{100 \times \sum C_{N_{recognised}}}{C_N}, \; z = \frac{\sum C_{N_{recognised}}}{M_N} \quad (16)$$

where y is the recognition accuracy (locally for each subject or action), estimated by at least one ensemble program, and z is the confidence probability assessing the recognition validity. The recognition validity assesses the overall number of votes acquired from all the ensemble programs. Further, the behaviour recognition has been estimated by averaging the recognition of two actions that belong to the same behaviour. In all, from the average values of Table 2(a), the best performed subject was sub3, and the worst sub4. Also, the action with the highest recognition was the side-kicking one, which spotlighted the Lower behaviour to be the most prominent among the rest.

On the contrary, from the GM model's testing performance shown in Table 2(b), the best-of-run elitist classifier has achieved 56.6% of recognition accuracy, utilising a single model. Recall that the classical architecture of the GM model uses the joint probability weighting to enhance the recognition performance. However, the best accuracy was not given by an ensemble of programs; actually, the ensemble programs induced lower accuracies than the single model. Corollarily, there is a 17% difference in accuracy between the GGPA and the GM. Significant difference is also observed on the average accuracies, as well as on the behaviour recognition. Thereupon, the training similarities observed in Section 4.1 have been proven correct by the testing performance as well. In other words, both methodologies tackled the action recognition problem with very similar training error, and testing accuracy, with respect to the evaluation of a single model. The GGPA model has achieved better recognition accuracy when only the SV ensemble method was introduced.

A final issue is to see the evolved complexity that has been adopted by either of the two models. Figs. 6(a), 6(b), and 6(c) picturise the structural complexity of the GGPA models. The tree expressions, as seen in the bloating paths of Fig. 4, have kept small depths (average ~ 5), and sizes (average ~ 75 primitives). In addition, it is observed a symmetry on the way the expressions have evolved and spread their branched. Overall, the GGPA models have remained small and simple, which is a crucial factor to be further interpreted. The GM model, shown by the expression of Fig. 6(d), has evolved a grown (asymmetric) individual that has reached the maximum depth of 17, along with a couple of hundreds of primitives. In a closer look of both groups



(a) (b) (c) (d)

Figure 6: Structural complexity of the classifier expressions. (a, b, c) The GGPA ensemble programs (program indices: 0, 2, and 9 respectively), (d) The GM program.

of expressions, it is seen that the GGPA model harnesses simple genotypic structures to induce recognition. This is distinctive only when appropriate ensemble methods are introduced such as the SV.

5. CONCLUSIONS

In this paper, an innovative GP fitness measure was introduced, designated for the recognition of six aggressive physical activities. For the first time in the literature of evolutionary feature selection, a set of biomechanical primitives has been employed to model physical activities for action and behaviour recognition purposes. The proposed evolutionary architecture harnesses a biomechanical primitive language that comprises features from *kinematics*, *dynamics*, and *energetics* categories (Table 1). The primitives have been employed to evolve solutions for the N-class problem of three aggressive behaviours and six actions. The guidance of the evolutionary process was accomplished by the induction of an novel probabilistic fitness measure, which represents Gaussians of program evaluations with geometric circles (Section 2). The circle representation is manifested in a ground-plan projection, by associating the probabilistic properties of a Gaussian distribution with the geometric properties of a circle.

The proposed method (GGPA), was experimentally compared with the classical method (GM). From the comparison, the models training and the testing performances were examined. Satisfactory results have been obtained by the testing recognition performance of the GGPA model, with maximum classification accuracy reaching the 73.3%, whilst the GM model derived with poorer recognition, 56.6%. However, it has been found that both models are equally powerful, with the difference in the weighting methods used to ensemble optimal solutions. In fact, the recognition accuracy of the GGPA model induced with the use of the *Statistical Voting* method, and the ensemble of three programs. Instead, the GM model did not make use of an ensemble of programs to achieve high recognition.

References

[1] W. Smart and Zhang M. Probability based genetic programming for multiclass object classification. Technical report, Computer Science, Victoria University of Wellington, 2004.

[2] W. D. Smart and M. Zhang. Using genetic programming for multiclass classification by simultaneously solving component binary classification problems. In *EuroGP*, pages 227–239, 2005.

[3] T. Theodoridis, P. Theodorakopoulos, and H. Hu. Evolving aggressive biomechanical models with genetic programming. In *Proc. of the 2010 IEEE/RSJ Int. Con. on Intelligent Robots and Systems*, pages 2495–2500, 2010.

[4] M. Zhang and W. D. Smart. Multiclass object classification using genetic programming. In *EvoWorkshops*, pages 369–378, 2004.

[5] W. B. Langdon and B. F. Buxton. Genetic programming for combining classifiers. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001*, pages 66–73. Morgan Kaufmann, 2001.

[6] L. Rokach and O. Maimon. *Data Mining with Decision Trees*, volume 69. World Scientific Publishing Co. Pte. Ltd., 2008.

[7] Thompson and Kelvin. Area of intersection: two circles. *Graphics gems, Academic Press Professional, Inc.*, pages 43–46, 1990.

[8] J. R. Koza. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[9] K. Chellapilla. Evolving computer programs without subtree crossover. *IEEE Trans. on Evolutionary Computation*, 1:209–216, 1997.

[10] R. Poli and W. B. Langdon. Backward-chaining evolutionary algorithms. *Artificial Intelligence*, 170(11):953–982, 2006.

[11] Code of ethics and conduct. Technical report, The British Psychological Society, 2006.

[12] T. J. Walilko, D. C. Viano, and C. A. Bir. Biomechanics of the head for olympic boxer punches to the face. *British Journal of Sports Medicine, ProBiomechanics LLC*, 39:710–719, 2005.

[13] E. K. Burke, S. Gustafson, and G. Kendall. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004.