33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Improving Parkinson's Disease Identification Through Evolutionary-Based Feature Selection

André A. Spadoto, Rodrigo C. Guido, Felipe L. Carnevali, André F. Pagnin, Alexandre X. Falcão, João P. Papa

Abstract— **Parkinson's disease (PD) automatic identification has been actively pursued over several works in the literature. In this paper, we deal with this problem by applying evolutionary-based techniques in order to find the subset of features that maximize the accuracy of the Optimum-Path Forest (OPF) classifier. The reason for the choice of this classifier relies on its fast training phase, given that each possible solution to be optimized is guided by the OPF accuracy. We also show results that improved other ones recently obtained in the context of PD automatic identification.**

I. Introduction

Parkinson's disease (PD) automatic identification has been extensively studied in the last years, and much effort has been dedicated to find the features that really matter to this task. Navío et al. [1] presented a collection of feature selection algorithms to deal with PD recognition, and Little [2] developed a remarkable study about PD identification and also introduced a new feature to improve the effectiveness of such systems. Revett et al. [3] introduced the Rough sets theory for feature selection in the context of PD automatic recognition. Finally, Guo et al. [4] have introduced a learning function composed by a Gaussian Mixture Model and Genetic Programming in order to accomplish this task.

A particular attention has been devoted to evolutionary-based algorithms for feature selection, since to find a subset of features that maximizes the accuracy of a classifier can be seen as an optimization procedure. In this paper, we improved the results presented by Spadotto et al. [5], which introduced the Optimum-Path Forest (OPF) [6] classifier in the context of PD automatic identification, by applying three recent algorithms for feature selection based on evolutionary intelligence: Particle Swarm Optimization (PSO) [7], Harmony Search (HS) [8] and Gravitational Search Algorithm - GSA [9]. Our results are comparable to the ones obtained by Little et al. [2], and the optimal subset of features can be calculated in less than 1 second using the Oxford Parkinson's Disease Detection Dataset [2].

This work was Partially supported by CNPq Grants # 481556/2009-5 (ARPIS) and #303673/2010-9, and FAPESP Grant # 2009/16206-1.

A. Spadotto and Rodrigo Guido are with Institute of Physics at São Carlos, University of São Paulo, São Carlos, Brazil spadotto@gmail.com, guido@ifsc.usp.br

F. Carnaveli is with Department of Computing, Federal University of São Carlos, São Carlos, Brazil, felipe.carnevali@gmail.com

A. X. Falcão is with Institute of Computing, University of Campinas, Campinas, Brazil, afalcao@ic.unicamp.br

A. Pagnin and J. Papa are with Department of Computing, UNESP - Univ Estadual Paulista, Bauru, Brazil, afpagnin@gmail.com, papa@fc.unesp.br

The remainder of this paper is organized as follows. Sections II and III describe, respectively, the evolutionary-based feature selection theory, the dataset and methodologies used in this work. Section IV address the experiments. Finally, Section V states conclusions.

II. Evolutionary-based feature selection

This section reviews the evolutionary-based techniques for feature selection used in this work: PSO, HS and GSA.

A. Particle Swarm Optimization

Basically, the Particle Swarm Optimization - PSO is a technique modeled on swarm intelligence that finds a solution in a search space based on the social behavior dynamics [7]. Each possible solution of the problem is modeled as a particle in the swarm that imitates its neighborhood based on a fitness function. In this context, each particle has a memory that stores its best local solution (local maxima) and the best global solution (global maxima).

The entire swarm is modeled in a multidimensional space $\Re^m$, in which each particle $p_i = (x_i, v_i) \in \Re^m$, $i = 1, 2, \ldots, n$, has two main features: (i) position (x_i) and (ii) velocity (v_i). The local (best current position $\widehat{x_i}$) and global solution $\widehat{s}$ are also known. After defining the swarm size, i.e., the number of particles, each one of them is initialized with random values of both velocity and position. Each individual is then evaluated with respect to some fitness function and its local maximum is updated. At the end, the global maximum is updated with the particle that achieved the best position into the swarm. This process is repeated until some convergence criterion is reached. The updated position and velocity equations of the particle p_i in the simplest form that govern the PSO are, respectively, given by

$$v_i = wv_i + c_1 r_1(\widehat{x}_i - x_i) + c_2 r_2(\widehat{s} - x_i) \quad (1)$$

and

$$x_i = x_i + v_i, \quad (2)$$

where w is the inertia weight that controls the interaction power between particles, and $r_1, r_2 \in [0, 1]$ are random variables that give the idea of stochasticity to the PSO method. Constants c_1 and c_2 are used to guide particles into good directions.

Recently, Ramos et al. [10] presented a hybrid algorithm for feature selection based on PSO and OPF (PSO-OPF), in

which the main idea was to use the accuracy of OPF over an evaluating set as the fitness value for PSO. Given that the Optimum-Path Forest classifier has been demonstrated to be accurate and very fast for training patterns [6], the combination of PSO and OPF can provide a fast and robust solution for feature selection, specially in large datasets. The PSO-OPF algorithm uses the accuracy of OPF to guide PSO to find a suitable solution. Basically, for each particle, the OPF classifier is trained with the correspondent subset of features in a training set, and its accuracy is assessed over an evaluating set. More details can be found in [10].

B. Harmony Search

The Harmony Search (HS) is an evolutionary algorithm inspired in the music, considering the improvisation process of music players [8]. The main idea is to use the same process adopted by musicians to create new songs to obtain a near-optimal solution of some optimization process. Basically, any possible solution is modeled as a harmony and each parameter to be optimized can be seen as a musical note. The best harmony (solution) is chosen as the one that maximizes some optimization criteria. The algorithm is composed by few steps, as described in the next sections.

1) The Optimization Problem and Algorithm Parameters: In order to describe how HS works, an optimization problem is specified in Step 1 as follows:

$$Minimize\ f(x)\ subject\ to\ x_i,\ \forall i = 1, 2, \ldots, n, \quad (3)$$

where $f(x)$ is the objective function, $x_i \in X$ means the harmony i and n is the size of X, i.e., the set of all harmonies.

The HS algorithm parameters required to solve the optimization problem (Equation 3) are also specified in this step: the harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and stopping criteria. HMCR and PAR are parameters used to improve the solution vector, i.e., they can help the algorithm to find globally and locally improved solutions in the harmony search process (Step 3). Recall that in this paper we set $n = HMCR$.

2) Harmony Memory (HM): Now, let us define x_i^j as the j-th value of harmony i. In Step 2, the HM matrix (Equation 4) is initialized with randomly generated solution vectors with their respective values of the objective function:

$$HM = \left[\begin{array}{cccc|c} x_1^1 & x_1^2 & \ldots & x_1^m & f(x_1) \\ x_2^1 & x_2^2 & \ldots & x_2^m & f(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \ldots & x_n^m & f(x_n) \end{array}\right]. \quad (4)$$

3) Generating a New Harmony From HM: In Step 3, a new harmony vector $x_i^{'}$ is generated from the HM based on memory considerations, pitch adjustments, and randomization (music improvisation). It is also possible to choose the new value using the HMCR parameter, which varies between 0 and 1 as follows:

$$x_i' \leftarrow \begin{cases} x_i' \in \{x_i^1, x_i^2, \ldots, x_i^m\} & \text{with probability HCMR,} \\ x_i' \notin X_i & \text{with probability (1-HCMR).} \end{cases} \quad (5)$$

The HMCR is the probability of choosing one value from the historic values stored in the HM, and (1- HMCR) is the probability of randomly choosing one feasible value not limited to those stored in the HM.

Further, every component of the new harmony vector $x_i^{'}$ is examined to determine whether it should be pitch-adjusted:

$$\text{Pitching adjusting decision for } x_i' \leftarrow \begin{cases} \textbf{Yes} \text{ with probability PAR,} \\ \textbf{No} \text{ with probability (1-PAR).} \end{cases} \quad (6)$$

The pitch adjustment of each instrument is often used to improve the solutions and to escape from local optima. This mechanism concerns with shifting the neighboring values of some decision variable in the harmony. If the pitch adjustment decision for the decision variable x_1' is Yes, x_1' is replaced as follows:

$$x_i' \leftarrow x_i' + rb, \quad (7)$$

where b is an arbitrary distance bandwidth for the continuous design variable, and r is a uniform distribution between 0 and 1.

4) Update HM: In Step 4, if the new harmony vector is better than the worst harmony in the HM, the latter is replaced by this new harmony.

5) Stopping Criteria: In Step 5, the HS algorithm finishes when it satisfies the stopping criteria. Otherwise, Steps 3 and 4 are repeated in order to improvise a new harmony again.

The HS algorithm is an interesting approach for several applications, mainly because of its simplicity and low computational cost. Ramos et al. [11] proposed a Harmony Search-based feature selection algorithm, in which the idea is the same as PSO-OPF, i.e., to use the OPF accuracy over an evaluation set as the fitness function. This approach is called HS-OPF. Basically, for each harmony, an OPF classifier is trained with the correspondent subset of features in a training set, and its accuracy is assessed over an evaluating set. Thus, for each iteration of the HS algorithm, the harmonies that compose the HM are tunned, in order to provide the best accuracy over the evaluating set.

C. Gravitational Search Algorithm

Gravity is one of the four fundamental interactions of nature, along with the strong force, electromagnetism and the weak force. The idea that rules gravity concerns with the fact that an object with mass attracts one another. One of the most accepted theory is the Newton's law of universal gravitation, which says that "every massive particle in the universe attracts other massive one with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them":

$$F = G\frac{M_1 M_2}{R^2}, \quad (8)$$

in which M_1 and M_2 are the masses of particles 1 and 2, respectively, R^2 is the distance between them, G is a gravitational constant and F is the magnitude of the gravitational force.

The "gravitational constant" G is time dependent, and decreases with the age of universe:

$$G(t) = G(t_0)\frac{t_0}{t}^{\beta}, \ \beta < 1, \tag{9}$$

in which $G(t)$ is the value of gravitational constant at time t, and $G(t_0)$ is the value of the gravitational constant at the time of the "creation of the universe" that is being considered.

Newton's second law says that when a force F is applied to a mass, its acceleration a only depends on the force and its mass M:

$$a = \frac{F}{M}. \tag{10}$$

Based on above definitions, Rashedi et al. [9] proposed the Gravitational Search Algorithm (GSA), which can be defined as follows. Let $X = \{x_1, x_2, \ldots, x_n\}$ be an universe with n masses, such that $x_i \in \Re^m$. One can define, at a specific time t, the force acting on mass i from mass j in the dth dimension as following:

$$F_{ij}^d(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij}(t)+\epsilon}(x_j^d(t) - x_i^d(t)), \tag{11}$$

where $R_{ij}(t)$ is the Euclidean distance between masses i and j, and ϵ is a small constant.

In order to give a stochastic behavior to GSA, Rashedi et al. [9] assume the total force that acts on agent i in a dimension d as a randomly weighted sum of the forces exerted from other agents:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{n} \gamma_j F_{ij}^d(t), \tag{12}$$

in which γ_j denotes a randomly generated number between 0 and 1. The acceleration of mass i at time t and dimension d is given by:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \tag{13}$$

in which the mass M_i is calculated as follows:

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{n} q_j(t)}, \tag{14}$$

with

$$q_i(t) = \frac{f_i(t) - w(t)}{b(t) - w(t)}. \tag{15}$$

The terms $w(t)$ and $b(t)$ mean, respectively, the masses with worst and best fitness value. The term $f_i(t)$ denotes the fitness value of mass i.

Finally, to avoid local optimal solutions, only the best k masses, i.e., the ones with highest fitness values, will attract others. Let $\mathcal{K}$ be the set of these masses. The value of k is set to k_0 at the beginning of the algorithm and decreases with time. Hence, Equation 12 is rewritten as:

$$F_i^d(t) = \sum_{j\in\mathcal{K}, j\neq i} \gamma_j F_{ij}^d(t). \tag{16}$$

The velocity and position updating equations are given by:

$$v_i^d(t+1) = \gamma_j v_i^d(t) + a_i^d(t) \text{ and} \tag{17}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{18}$$

However, one can see that Equation 18 can not be applied in our case, since we are working in a multidimensional binary space. Therefore, Rashedi et al. [12] proposed the Binary GSA, which has the same formulation introduced above, but with a different equation for updating the position of each mass:

$$x_i^d(t+1) = \begin{cases} 1 - x_i^d(t) & \text{if } \gamma_i < S(v_i^d(t+1)) \\ x_i^d(t) & \text{otherwise,} \end{cases} \tag{19}$$

such that

$$S(v_i^d(t)) = \left|\tanh(v_i^d(t))\right|. \tag{20}$$

In order to achieve a good convergence rate, Rashedi et al. [12] proposed to limit the velocity to $\left|v_i^d\right| < v_{max} = 6$. According to them, this value was obtained over some experiments, and seemed to be appropriated in our case.

The GSA-based algorithm for feature selection works on a similar manner as the previous ones: the main idea is to use the OPF accuracy over an evaluating set as the fitness function to guide GSA onto searching the best solutions. This approach, called, GSA-OPF, was recently introduced by Papa et al. [13].

III. Materials and Methods

In this work, we have employed the Oxford Parkinson's Disease Detection Dataset [2], which is composed by biomedical voice measurements from 31 people, 23 with Parkinson's disease. Regarding features, 22 were extracted, as follows:

- MDVP:Fo - Average vocal fundamental frequency;
- MDVP:Fhi - Maximum vocal fundamental frequency;
- MDVP:Flo - Minimum vocal fundamental frequency ;
- MDVP:Jitter (%), MDVP:Jitter (Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP - Several measures of variation in fundamental frequency;
- MDVP:Shimmer, MDVP:Shimmer (dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA - Several measures of variation in amplitude;
- NHR, HNR - Two measures of ratio of noise to tonal components in the voice ;
- RPDE, D2 - Two nonlinear dynamical complexity measures;
- DFA - Signal fractal scaling exponent and
- Spread1, Spread2, PPE - Three nonlinear measures of fundamental frequency variation.

The dataset is composed by 193 samples, and there are around six recordings per patient. In this work, we compared

the effectiveness of OPF over the original dataset, i.e., without feature selection, with PSO-OPF, HS-OPF and GSA-OPF. The experiments have been executed 10 times with randomly generated training, evaluating and test sets. The percentages are: 30% for training, 20% to the evaluating set and the remaining 50% for testing. These percentage values were empirically chosen, based on our previous experience.

In regard to parameters, for all techniques we used 10 iterations for convergence and 300 initial solutions, i.e., particles for PSO, harmonies for HS and masses for GSA. With respect to PSO parameters, we used $c_1 = 1.4$, $c_2 = 0.6$ and $w = 0.4$, and for HS parameters we used $HMCR = 0.67$. Finally, for GSA parameters we used $G_0 = 1.4$ and $\epsilon = 0.6$. Notice that these values were empirically chosen, based on our previous experience.

IV. Experimental Results

The PD automatic identification using OPF has been addressed by Spadotto et al. [5]. In that case, the authors did not apply any feature selection technique. Thus, we repeated that experiments in order to compared their results against the ones obtained here. Table I displays the results. Note that we have used the evaluating set only for PSO-OPF, HS-OPF and GSA-OPF.

TABLE I
MEAN ACCURACY AND MEAN EXECUTION TIME TO SELECT THE MOST REPRESENTATIVE FEATURES IN SECONDS FOR PSO-OPF, HS-OPF AND GSA-OPF.

Technique	Accuracy	Feature selection execution time [s]	#features
OPF	71.16±5.44	-	22
PSO-OPF	73.53±8.18	1.54	14
HS-OPF	84.01±3.54	0.16	10
GSA-OPF	84.01±3.54	1.68	8

One can see that all feature selection techniques improved the results obtained by OPF over the original dataset. The most impressive results were obtained through HS-OPF and GSA-OPF techniques, in which the former selected the 10 out of 22 features: MDVP:Fo, MDVP:Jitter (%), MDVP:RAP, Jitter:DDP, MDVP:Shimmer (dB), MDVP:APQ, Shimmer:DDA, HNR, Spread2 and PPE (Section III). Recall that this information was obtained by randomly picking the output from one running. The HS-OPF technique was 10.5 times faster than GSA-OPF, which make it the best choice for the problem. The computational complexity of GSA is dominated by a sorting step in order to find the set of k masses that most influence positively the fitness value of a given one, as described by Equation 16. In our algorithm, this step was implemented with Quicksort [14].

Our results are not better than the ones reported by Revett et al. [3], since our accuracy measure [6] differs from them. If we have applied the same mathematical formulation as in [3], our recognition accuracy with HS-OPF would be around 92.78% with the above selected features, but still below of 100% obtained by them. However, we did not hear anything about their approach efficiency, since the number of rules generated by their proposed method may be a little bit prohibitive. In addition, some of the selected features by HS-OPF are identical to the ones obtained by Little et al. [2].

V. Conclusions

In this paper we deal with the problem of feature selection in the context of Parkinson's disease automatic identification using evolutionary-based techniques, in which the function to be maximized has been the one given by the OPF accuracy over an evaluating set.

The experiments have showed that all techniques improved the results over the traditional dataset, i.e., without feature selection. HS-OPF and GSA-OPF have achieved the best results, with HS-OPF selecting 10 out of 22 features and being the fastest approach. For future works, we intend to use meta-optimization techniques in order to select the parameters for PSO-OPF, HS-OPF and GSA-OPF.

References

[1] M. Navío, J. J. Aguilera, M. J. del Jesus, R. González, Francisco Herrera, and C. Iríbar, "Feature selection algorithms applied to parkinson's disease," in *Proceedings of the Second International Symposium on Medical Data Analysis*, London, UK, 2001, ISMDA '01, pp. 195–200, Springer-Verlag.

[2] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, "Suitability of dysphonia measurements for telemonitoring of parkinson's disease," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1015–1022, 2009.

[3] K. Revett, F. Gorunescu, and A.-B. M. Salem, "Feature selection in parkinson's disease: A rough sets approach," in *Proceedings of the International Multiconference on Computer Science and Information Technology*, Mragowo, Poland, 2009, pp. 425–428.

[4] P.-F. Guo, P. Bhattacharya, and N. Kharma, "Advances in detecting parkinsons disease," *Medical Biometrics*, vol. 6165/2010, pp. 306–314, 2010.

[5] A. A. Spadotto, J. P. Papa R. C. Guido, and A. X. Falcão, ," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, 2010, pp. 6087–6090.

[6] J. P. Papa, A. X. Falcão, and Celso T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[7] J. Kennedy and R.C. Eberhart, *Swarm Intelligence*, M. Kaufman, 2001.

[8] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, Springer Publishing Company, Incorporated, 1st edition, 2009.

[9] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[10] C. C. O Ramos, J. P. Papa, A. N. Souza, and A. X. Falcão, "What is the importance of selecting features for non-technical losses identification?," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, 2011, (accepted for publication).

[11] C. C. O Ramos, A. N. Souza, and J. P. Papa, "New insights on non-technical losses characterization through evolutionary-based feature selection," in *IEEE Transactions on Power Delivery*, 2011, submitted.

[12] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi, "Bgsa: binary gravitational search algorithm," *Natural Computing*, vol. 9, pp. 727–745, 2010.

[13] J. P. Papa, A. F. Pagnin, S. A. Schellini, A. A. Spadotto, R. C. Guido, M. P. Ponti Jr., G. Chiachia, and A. X. Falcão, "Feature selection through gravitational search algorithm," in *Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, 2011, (accepted for publication).

[14] C. A. R. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–15, 1962.