The Enhancement of Speech Recognition Using Particle Swarm Optimization

Virtyt Lesha
Department of electronics and telecommunications
Polytechnic University of Tirana
Shkoder,Albania
virtytl@hotmail.com

***Abstract* - The identification of sound is a process that consists in several sub process attached to each other and that are executed according to a well-defined order. Identifying rough voice is finding through a certain algorithm to determine which of the actual sounds recorded are reference.**

To achieve the desired results the signal, at the entrance of the system, needs to pass in the process of conversion from analogue signal to digital signal.

In computer science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but, is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

The PSO algorithm starts by generating random positions for the particles, within an initialization. Velocities can also be initialized to zero or to small random values to prevent particles from leaving the search space during the first iterations. During the main loop of the algorithm, the velocities and positions of the particles are iteratively updated until a stopping criterion is met.

Three terms in the velocity-update rule above characterize the local behavior that particles follow. The first term, called the inertia or momentum serves as a memory of the previous flight direction, preventing the particle from drastically changing direction. The second term, called the cognitive component models the tendency of particles to return to previously found best positions. The third term, called the social component quantifies the performance of a particle relative to its neighbors. It represents a group norm or standard that should be attained.

In some cases, particles can be attracted to regions outside the feasible search space. For this reason, mechanisms for preserving solution feasibility and a proper swarm operation have been devised. One of the least disruptive mechanisms for preserving feasibility is one in which particles going outside, are not allowed to improve their personal best position so that they are attracted back to the feasible space in subsequent iterations.

Actually, there is not a specific mathematical function that connects the ID of the sounds and the time, expressed in seconds, to execute the PSO algorithms. Therefore to find the connection between them, statistical analysis is needed. The methodology consists in putting as a position variable of sound's ID against the parameters of time needed to execute the standard PSO algorithm as well as the enhanced one. Through Matlab, these data are analyzed through polynomial regression analysis giving the determination coefficients. Also for credibility of the results we have simulated in Matlab even the ANOVA table and the Pearson coefficients.

Finally, we refer to a standard time needed to perform the speech recognition using PSO algorithm and the finally we prove that the time needed to recognize the speech take using the improved PSO algorithms about 30% less that the standard one.

Keywords –PSO, speech, Matlab, ANOVA, Pearson

I. INTRODUCTION

The identification of sound is an interdisciplinary field of linguistics that makes possible the inclusion of knowledge and research in linguistics, computer science and electronics engineering to develop methods and technologies that enable the identification and translation of the spoken language in texts from the computer and other smart equipments. This technology is also known as automatic speech recognition, speech recognition, etc.[6]

Some of voice identification systems use training strategies, where a person reads the text in the system. Then, the system analyzes the person's specific voice and uses it to tune the conversation identity of the person, thus resulting in increased

accuracy.[8] Systems that do not use training are called "independent systems", and those which use such training systems are called “dependent systems”. [5]
Voice identification applications include voice user interface such as call routing, search, simple data entry, preparation of structured documents, conversation processing, etc.
The term “voice recognition” or “speaker recognition” refers to the identification of voice, and not what is said. The identification of sound can simplify the task of translating the conversation in systems that are prepared in the voice of a specific person or can be used to authenticate or verify the identity of a speaker as part of a security process.
From the perspective of technology, identifying the voice has a long history with different waves of innovation. More recently, this research area has benefited from developments in deep learning and big data. Developments are evidenced not only by the publication of scientific articles published about this field, but what is more important, by adapting the industry worldwide to deep learning methods in the design of voice identification systems. Such industries include Microsoft, Google, IBM, Apple, Amazon, etch where many of them have published the core technology in their systems to detect sound based on deep learning.
Particle swarm optimization consists of a calculation method that optimizes a problem trying to iteratively improve a solution in relation to a specific parameter of quality. PSO optimizes a problem having a population of candidate solutions, which are called “dubbed”, by moving particles and these particles around a search space based on simple mathematical formulas in the position and velocity of the particles. Every movement of particles is influenced by the positioning of her best but she is guided towards the best position in space research, which may be updated as the best position. This makes the swarm to move toward better solutions.
PSO is metaheuristic while doing little or no assumptions about the problem that is optimized and can require very large spaces candidate solutions. However, metaheursitics as PSO does not guarantee an optimal solution. Specifically, PSO does not use the gradient of the problem that is optimized, which means that the PSO does not require that the problem of optimization to be differentiated as required by classical optimization methods such as gradient descent and quasi-Newton methods. Therefore, PSO can be used in optimization problems that are partly irregular, variable over time, etc.[4]

II. METHODOLOGY

In this study we have designed a simulation design through Matlab that minimizes the time execution of speech recognition through particle swarm optimization. [3]
Concretely, we used 100 sounds from an open source sound dataset and we have attached it in the simulation program of standard particle swarm optimization algorithm. This dataset contains different types of sounds such as: noise environmental sounds, urban sounds, human speech, etc. After creating the sound database in the program we have retrieved the output which consists in the time execution of each sound of the database in question. Then, in the same way we have followed this procedure with the improved version of the program, thus receiving the same output. From these outputs, we expect that the time execution of the improved particle swarm optimization algorithm will be around 30% less than the standard algorithm.
These results will be subject of statistical analysis. First, we will perform the polynomial non-linear regression analysis of sixth degree for the two outputs. The reason for the use of the nonlinear regression of this order is related to the fact that the use of nonlinear regression of higher orders does not represent more accurate apparent than nonlinear regression of higher orders. On the other side the regressions with lower orders has not been used because their accuracy is significantly lower than the order of regression in question. Second, to confirm the credibility of the results we will calculate the ANOVA (analysis of variance) table, thus interpreting the results. Finally, we will calculate the Pearson correlation coefficients to study the level of correlation between the time execution of standard particle swarm optimization algorithm and the improved one.

III. THE RESULTS OF THE SIMULATIONS

In this section we will observe the statistical processing of the data correlation between the standard PSO approach and the improved one. We have used statistical methods of the data processing which provide a continuity trend of these data and this trend is reflected in equations of line, in tables and coefficients of determination or correlation. These coefficients are an indicator that shows the order of dependency of data from each other. The statistical data processing are performed through Matlab.[1]
Let us treat the polynomial regression of sixth order of two relationships:

- The relationship of the time needed to recognize the speech with the standard PSO algorithm
- The relationship of the time needed to recognize the speech with the improved PSO algorithm

In the figure below we have shown the dependency of execution time of the standard PSO algorithm, as well the respective polynomial regression curves of sixth order:

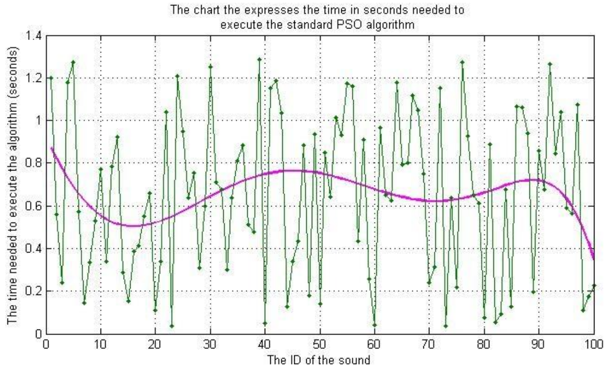

Fig. 1. The graphic results that express the continuity of the execution time of the standard PSO algorithms as well as the respective trend-line

This graph shows the case of the performance of the standard PSO on speech recognition. As we can see, there are two lines. The green line expresses the continuity of the execution time of speech recognition of every of 100 testing sounds. Also, the purple line expresses the non-linear polynomial regression curve of sixth order of this trend-line. The coefficient of determination of this trend-line is r^2=0.2883 that means that the dependency is 28.83% and the rest remains to be studied.

Let us see the figure 2:

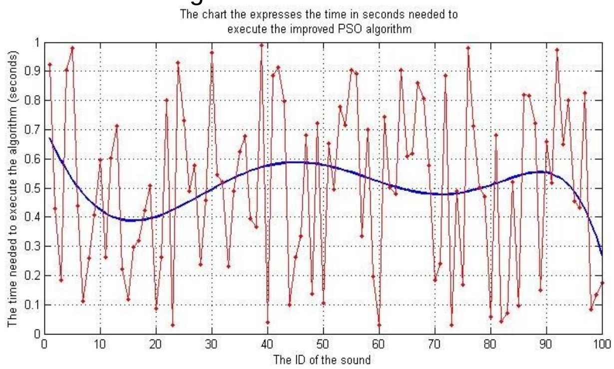

Fig. 2. The graphic results that express the continuity of the execution time of the improved PSO algorithms as well as the respective trend-line

This graph shows the case of the performance of the improved PSO on speech recognition. As we can see, there are two lines. The red line expresses the continuity of the execution time of speech recognition of every of 100 testing sounds. Also, the blue line expresses the non-linear polynomial regression curve of sixth order of this trend-line. The coefficient of determination of this trend-line is r^2=0.3748 that means that the dependency is 37.48% and the rest remains to be studied.

An important fact is that, as we can see from the figure, the shape of the trend-lines of the respective charts, are approximately equal. This is evident because the improved PSO algorithm has a performance of time execution approximately 30% smaller than the standard PSO algorithm regarding our speech recognition process. This means that there is linear relationship between these matrices of these time executions and thus there is this kind of graphs in the figure mentioned above.

Analysis of variance or ANOVA is a methodology that enables us to compare the means of different groups, so it might be more informative to call it the ***analysis of variation about means.*** There are many different types of ANOVA that can be performed in Matlab, but we have treated only one of them in This is called the ***one-way analysis of variance***.

When we calculate the ANOVA table two figure windows are opened, as shown in the figure below

One window contains side-by-side notched box-plots for the groups. This provides several things. First, it allows us to assess whether our assumptions of normality and equal variance are reasonable, which they seem to be for this example. Second, the notched box-plots give us a visual way to test whether the *medians* are different. If the intervals given by the notches do not overlap, then we have evidence that the *medians* are different.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	0.912	1	0.91203	7.23	0.0078
Error	24.974	198	0.12613		
Total	25.8861	199			

Fig. 3. a. The output of implementing the one-way ANOVA

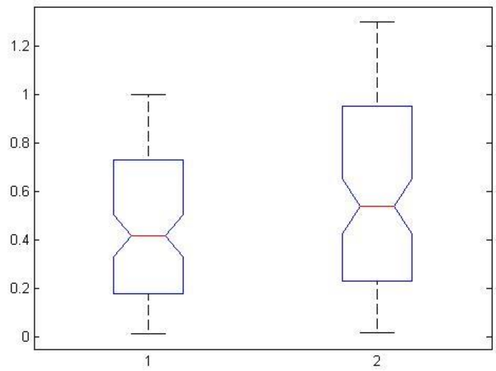

Fig. 3. b. The output of implementing the one-way ANOVA

This is the graphical output from **anova1**. The sepal width was used for this analysis, and we see a notched box-plot of the two groups. There appears to be significant evidence that the **medians** between group 1 and group 2 are different because the notched intervals do not overlap. The results of the ANOVA test are shown in the table. The second column is the sum of squares (SS); the third column is the degrees of freedom; and the fourth column is the mean of squares (SS/df). The observed value of the F-statistic and the corresponding p-value are also shown. The p-value is very small, so we have evidence that the pair of group means is significantly different.

The alternative hypothesis in a one-way ANOVA is that at least one pair of group means is significantly different. However, what pair is different? We can use the **multiple comparison test** in the Matlab Statistics Toolbox to determine which pair is different. Below we have shown the results of this test.

1.0000 2.0000 -0.2335 -0.1351 -0.0366

The first two elements of this output represent the group numbers. The third and fifth elements are the end points of a 95% confidence interval for the *difference of the group means*, and the fourth element is the estimated difference. Thus, the difference in the means is **-0.1351**, and the confidence interval for the difference is given by

[-0.2335 , -0.0366].

The interval does not contain zero, so we can conclude that this pair of means is significantly different.

We also get some other helpful graphical output from the **multcompare** function. Matlab automatically opens a window, similar to what is shown in the figure below. There will be a horizontal line for each group, where the line is a graphical representation of the estimated mean and a 95% comparison interval for the mean. We can click on each of the lines in the plot, and MATLAB will display the result of the multiple comparison tests. Groups with significantly different means will be shown in red.[9]

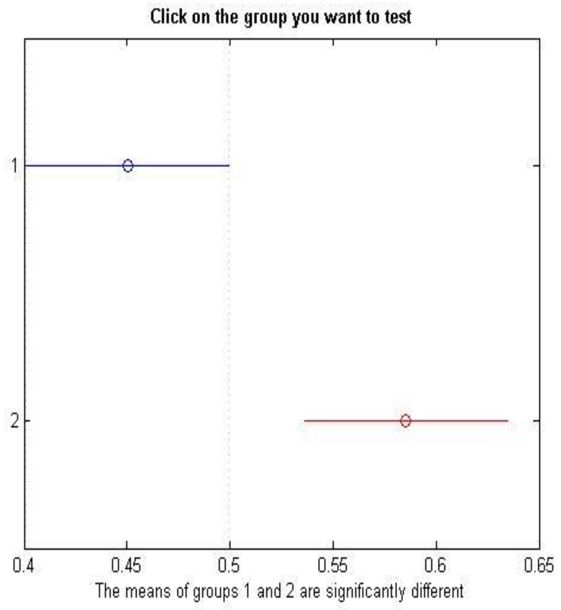

Fig 4. a – The output of implementing the multiple comparison tests

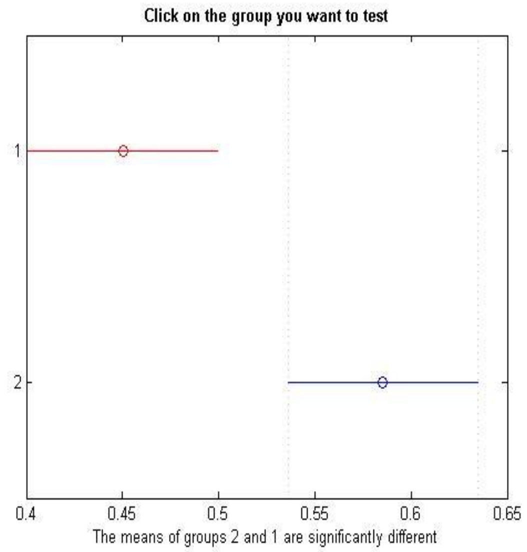

Fig 4. b. The output of implementing the multiple comparison tests

This is the graphical output from the **multcompare** function. There is one line per group. The circle represents the estimated mean for the group, and the line is the 95% comparison interval for the estimated mean. We can click on each of the lines to find out what other groups are significantly different from the one selected.

Often several quantitative variables are measured on each member of a sample. If we consider a pair of such variables, it is frequently of interest to establish if there is a relationship between the two; i.e. to see if they are *correlated*.

We can categorize the type of correlation by considering as one variable increases what happens to the other variable:

- *Positive correlation* – the other variable has a tendency to also increase;
- *Negative correlation* – the other variable has a tendency to decrease;
- *No correlation* – the other variable does not tend to either increase or decrease.

Pearson's correlation coefficient is a statistical measure of the strength of a *linear* relationship between paired data. In a sample it is denoted by *r* and is by design constrained as follows

$$-1 \le r \le 1$$

Furthermore:

- Positive values denote positive linear correlation;
- Negative values denote negative linear correlation;
- A value of 0 denotes no linear correlation;
- The closer the value is to 1 or –1, the stronger the linear correlation.

In our case, from the Matlab programming, the Pearson coefficient resulted to be **0.8934.** This means that the correlation is very strong and this evidence, as we cited above, come from the fact the the relationship between the time execution of standard PSO algorithm and the improved one, is approximately linear. [2]

IV. CONCLUSIONS

In this article we gave the results of the simulation via Matlab of the performance of improved PSO algorithm regarding speech recognition. For this objective we have utilized 100 sounds from an open source sound dataset and we have attached it in the simulation program of standard particle swarm optimization algorithm. This dataset contains different types of sounds such as: noise environmental sounds, urban sounds, human speech, etc.

These results have been subject of statistical analysis. First, we performed the polynomial non-linear regression analysis of sixth degree for the two outputs. The reason for the use of the nonlinear regression of this order was related to the fact that the use of nonlinear regression of higher orders does not represent more accurate apparent than nonlinear regression of higher orders. On the other side the regressions with lower orders has not been used because their accuracy is significantly lower than the order of regression in question. In the case of the performance of the standard PSO on speech recognition, we have shown the execution time of speech recognition of every of 100 testing sounds. Also, we have shown the trend-line of the non-linear polynomial regression curve of sixth order. The coefficient of determination of this trend-line was $r^2=0.2883$ that means that the dependency is 28.83% and the rest remains to be studied. Furthermore, in the case of the performance of the improved PSO on speech recognition we have shown the execution time of speech recognition of every of 100 testing sounds. Also, we have shown the trend-line of the non-linear polynomial regression curve of sixth order. The coefficient of determination of this trend-line was $r^2=0.3748$ that means that the dependency is 37.48% and the rest remains to be studied.[7]

Second, to confirm the credibility of the results we calculated the ANOVA (analysis of variance) table, we interpreted the results. Finally, we calculated the Pearson correlation coefficients to study the level of correlation between the time execution of standard particle swarm optimization algorithm and the improved one. From the simulation of ANOVA table, we confirm that the sepal width used for this analysis consisted in a notched box-plot of the two groups. [10] There appeared to be significant evidence that the **medians** between group 1 and group 2 were different because the notched intervals did not overlap. The p-value was very small, so we have evidence that the pair of group means is significantly different. Also, we know that the alternative hypothesis in a one-way ANOVA wass that at least one pair of group means is significantly different. We used the **multiple comparison test** in the Matlab Statistics Toolbox to determine which pair is different. The confidence interval did not contain zero, so we can conclude that this pair of means is significantly different.

Third, from the Matlab programming, the Pearson coefficient resulted to be **0.8934.** This means that the correlation is very strong and this evidence, as we cited above, come from the fact that the relationship between the time execution of standard PSO algorithm and the improved one, is approximately linear.

The limitations of this paper are related to the fact that we have used the nonlinear regression method of the sixth order to give the tendency of these data which may lead to a prediction of this dependency. The reason for the use of the nonlinear regression of this order is related to the fact that the use of nonlinear regression of higher orders does not represent more accurate apparent than nonlinear regression of higher orders. On the other side the regressions with lower orders has not been used because their accuracy is significantly lower than the order of regression in question.

Finally I propose as future work the use of nonlinear regression of higher degrees, such as affecting the results so that to show more correct approximations to these kind of parameters.

V. REFERENCES

[1] Boudreaux-Bartels, G. F. and Murry, R. Time-frequency signal representations for biomedical signals. In: The Biomedical Engineering Handbook. Piscataway, N.J., 1995.

[2] Bruce, E. N. Biomedical Signal Processing and Signal Modeling, John Wiley and Sons, New York, 2001.

[3] Chaparro., L, “Signals and systems”, 2015, ELSEVIER, pp.78

[4] Clerc., M, “Particle Swarm Optimization”, 2013, ISTE”

[5] Mikki., S, Kishk., A, “Particle Swarm Optimization – a physics-based approach , ”Morgan and Claypool Publishers”

[6] Shi., Yuhui. “Empirical study of particle swarm optimization”, IEEE Xplore

[7] Sonka, M., Hlavac V., and Boyle R. Image processing, analysis, and machine vision. Chapman and Hall Computing, London, 1993.

[8] Stearns, S.D. and David, R.A Signal Processing Algorithms in MATLAB, Prentice Hall, Upper Saddle River, NJ, 1996.

[9] Trelea., I, “The particle swarm optimization algorithm: convergence analysis and parameter selection”, ELSEVIER

[10] https://serv.cusp.nyu.edu/projects/urbansounddataset/download-urbansound.html