International Journal of Applied Mathematics and Computation
Vol. 5 No. 1 (June, 2019)

KALAHARI JOURNALS

Received: 13th August 2018 Revised: 24th December 2018 Accepted: 10th March 2019

A brief look at the least-squares approach as a classifier applied to restricted-vocabulary speech recognition

Rodrigo Capobianco Guido[*], **Luciene Cavalcanti Rodrigues**

ABSTRACT

In this letter, a speech recognition algorithm based on the least-squares method is presented. Particularly, the intention is to exemplify how such a traditional numerical technique can be applied to solve a signal processing problem that is usually treated by using more elaborated formulations.

Keywords: least-squares method; speech recognition; "intelligent" transfer functions.

1. Introduction

In this letter, the potential offered by the least-squares method (LSM) [1] to solve incompatible systems of linear equations (ISLE) is explored in order to create a technique for speech recognition [2]. Preceded by a basic feature extraction step, LSM is used to solve an ISLE whose solution characterizes an "intelligent" transfer function (ITF) that transforms an n-dimensional input vector into a scalar value. According to this value, which corresponds to the algorithm output, it is possible to know which was the input spoken word.

The remainder of this work is organized as follows. A very brief review on LSM, plus some concepts related to speech recognition, are presented in section 2. Based on such information, section 3 describes the proposed approach, then, section 4 presents tests and results, and, lastly, section 5 describes the conclusions.

2. Brief review of important concepts

2.1 *The least-squares method to solve incompatible systems of linear equations*

In order to solve an incompatible linear system of M equations in N unknowns, being $M > N$, the LSM [1] is one of the possible, and most used, approaches. Its definition, which is well-documented in the literature [1], implies that, in practice, the generic linear system

$$\begin{cases} \alpha_{1,1}x_1 + \alpha_{2,1}x_2 + \alpha_{3,1}x_3 + ... + \alpha_{n,1}x_n &= \beta_1 \\ \alpha_{1,2}x_1 + \alpha_{2,2}x_2 + \alpha_{3,2}x_3 + ... + \alpha_{n,2}x_n &= \beta_2 \\ \alpha_{1,3}x_1 + \alpha_{2,3}x_2 + \alpha_{3,3}x_3 + ... + \alpha_{n,3}x_n &= \beta_3 \\ ... & . \ . \\ ... & . \ . \\ ... & . \ . \\ \alpha_{1,m}x_1 + \alpha_{2,m}x_2 + \alpha_{3,m}x_3 + ... + \alpha_{n,m}x_n &= \beta_m \end{cases},$$

[*] guido@ieee.org

which can be expressed in matricial form as being

$$\underbrace{\begin{pmatrix} \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \cdots & \alpha_{n,1} \\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \cdots & \alpha_{n,2} \\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \cdots & \alpha_{n,3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_{1,m} & \alpha_{2,m} & \alpha_{3,m} & \cdots & \alpha_{n,m} \end{pmatrix}}_{\text{matrix } A_{[\cdot][\cdot]}} \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}}_{\text{vector } X_{[\cdot]}} = \underbrace{\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \cdot \\ \cdot \\ \cdot \\ \beta_m \end{pmatrix}}_{\text{vector } B_{[\cdot]}} \quad ,$$

can be approximatelly solved by multipling its both sides by $A^T_{[\cdot][\cdot]}$, i.e., the transpose of $A_{[\cdot][\cdot]}$. We therefore have:

$$A^T_{[\cdot][\cdot]} \cdot A_{[\cdot][\cdot]} \cdot X_{[\cdot]} = A^T_{[\cdot][\cdot]} \cdot B_{[\cdot]} \quad ,$$

which corresponds to an N by N linear system. The exact solution of this last system is the best approximation, in the least-squares sense, to the incompatible original system. This methodology is adopted in this work as a part of the proposed approach.

2.2 *Speech recognition and feature extraction*

Speech recognition has been intensively studied during the last decades, as can be observed, for example, in [2]. Traditionally, speech recognition algorithms are based on Hidden Markov Models (HMMs) or neural networks [3], mainly, when there is a large set of speech utterances or words to be recognized. On the other hand, when a modest set of words is being considered, simpler procedures can be certainly adopted. This is the case explored in the current work.

Independent of the algorithm used to recognize the words, which require previous digitalization by a computer or a digital signal processing (DSP) hardware, there is always an n-dimensional vector, stored in the computer memory, that corresponds to the digitalized speech signal. The length of such a vector is not constant, being proportional to the length of the spoken word, to the time the speaker needs to sound such a word, and so on. Since each word corresponds to a vector of one particular dimension, i.e., with a particular length, the transfer function used to convert such a vector into a scalar value representing the spoken word should be adaptive, in order to deal with that fact. This adaptation is neither an easy task to be implemented nor is functional in practice.

Feature extraction is, most of the times, the technique used to solve the above-mentioned problem. This solution implies that a small set of P numerical parameters, or characteristics, P being a constant integer, is used to represent each spoken word. Thus, the role of a feature extraction step is to transform an n-dimensional vector, of variable length, into a considerable smaller vector with P parameters. Obviously, such parameters should be representative of the original spoken words. Many possibilities exist, being energies and fractal dimensions, for example, valid choices, according to [2] and [4].

3. The proposed approach

The description of the proposed approach follows, in the form of an algorithm. The values of W and T, mentioned in it, will be discussed later.

- **BEGINNING**
- **STEP** S_1**:** Define a small set of W words to be recognized. Then, by using a computer, a proper software configured to a sampling-rate of 8000 Hz - 16-bit, and a microphone, record T samples of each word. Include speakers of different ages, genders, and so, on in order to obtain a diversity of voices with statistical significance. Wave [5] is the most simple file format to save the recordings;

- **STEP** S_2**:** By using a computational tool, such as Matlab, read the wave files, importing their raw data, i.e., their digitalized signals values, into $V = W \cdot T$ vectors;

- **STEP** S_3**:** Normalize all the vectors so that they have zero mean and unitary standard deviations;

- **STEP** S_4**:** For each vector V_i, $(1 \leq i \leq (W \cdot T))$, extract a total of 15 parameters, P_1 to P_{15}, as follows:

  - P_1: the energy of the entire vector;
  - P_2 and P_3: the energies of the first and the second halfs of the vector, respectively;
  - P_4 to P_7: the energies of the first, second, third, and fourth forths of the vector, respectively;
  - P_8 to P_{15}: the energies of the first, second, ..., and eight eights of the vector, respectively;

  The use of 15 parameters is just the choice adopted here.

- **STEP** S_5**:** For each one of the W words, assign an integer value, q_j, to be used as a label;

- **STEP** S_6**:** In order to determine the proper weight, α_k, for each parameter, establish, for each $W \cdot T$ spoken word, one linear combination between α_k and the parameters P_i, i.e., a linear system so that:

$$\begin{pmatrix} P_{1,vector1} & P_{2,vector1} & \cdots & P_{15,vector1} \\ P_{1,vector2} & P_{2,vector2} & \cdots & P_{15,vector2} \\ P_{1,vector3} & P_{2,vector3} & \cdots & P_{15,vector3} \\ \cdots & \cdots & \cdots & \cdots \\ P_{1,vectorV_{(W \cdot T)}} & P_{2,vectorV_{(W \cdot T)}} & \cdots & P_{15,vectorV_{(W \cdot T)}} \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \cdots \\ \alpha_{15} \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \cdots \\ q_{(W \cdot T)} \end{pmatrix} \quad .$$

- **STEP** S_7**:** Since the system above is incompatible, apply the LSM to obtain an approximate solution, $\{\alpha_1, ..., \alpha_{15}\}$, to it;

- **STEP** S_8**:** In order to recognize a new unknown input word, apply the steps S_2 to S_4 to the corresponding wave file. This produces a set of fifteen parameters which are the features of the word. Then, calculate the value $R = \sum_{i=1}^{15} \alpha_k \cdot P_k$, which is the algorithm output. The label q_i that is closest to R corresponds to the unknown input word.
  END.

In the algorithm above, the sampling rate and the quantization chosen, i.e., at least 8000Hz, 16-bit, is the one necessary to preserve frequencies below 4000 Hz, according to Nyquist's sampling theorem [2]. This range of frequencies corresponds to the one in which voice signals are concentrated. In addition, the energy of a vector $Y[\cdot]$, i.e., $E(Y[\cdot])$, mentioned above, consists simply in the sum $\sum_{s=1}^{C} {y_s}^2$, being C the length of $Y[\cdot]$.

In relation to the values of W and T, the expectation is that the product $W \cdot T$, which corresponds to the number of equations in the linear system, should be considerable greater than the number of parameters P, that is equals to 7 in the proposed algorithm. This is because the intention is to create a transfer function that is built considering many examples of each word. In some specific case that $W \cdot T = P$, then, LSM would not be necessary to solve the system, however, there would certainly be a lack of examples to form a valid statistic. Finally, $W \cdot T < P$ is not considered.

4. Example tests and their corresponding results

As an example situation, $W = 4$ was adopted, being "down", "right", "up", and "left", the corresponding words, and 0, 1, 2, and 3, their corresponding labels, respectively. For each word, $T = 5$ utterances were collected from different speakers, including male, female, young, and elderly people. This totalizes $4 \cdot 5 = 20$ feature vectors of length fifteen each one. Carrying out the steps S_1 to S_7 of the algorithm presented in the previous section, the values of α_1 to α_{15} were determined.

In order to test the proposed technique, five additional utterances of each word were collected and the step S_8 of the algorithm was carried out. The results are listed in table 1, which has the format of a confusion matrix, in order to allow a more detailed description. As indicated in the table, the results

Table 1: results of the experiment.

input word / output result	"left"	"right"	"up"	"down"	TOTAL
"left"	**3**	**1**	**1**	**0**	**5**
"right"	**0**	**3**	**2**	**0**	**5**
"up"	**0**	**1**	**4**	**0**	**5**
"down"	**0**	**1**	**0**	**4**	**5**

can be considered positive, since only some errors were detected, being the accuracy, i.e., the number of correct results in relation to the total number of words tested, equals to $\frac{(3+3+4+4)}{(5+5+5+5)} = 0.7 = 70\%$. Researchers or students interested in repeating the experiment, can contact the author, by e-mail, in order to obtain the material used.

5. Conclusions

This paper presented an interesting application of the LSM for restricted-vocabulary speech recognition. Although the LSM is a well-known technique in numerical analysis, its application in speech processing had never been showed before in literature in a so simple way. The proposed approach illustrated how one can construct an "intelligent" transfer function, by using basic mathematical tools, which transforms an input feature vector into a numerical output, making possible to recognize certain spoken words.

Of fundamental importance is the fact that no comparison between the proposed technique and other approaches for restricted-vocabulary speech recognition was presented. This is due to the fact that this work does not intend to improve any other existing technique which shows a better accuracy. Instead, the only intention is, as discussed above, to illustrate how a basic numerical tool can be applied to solve a practical problem in the field of signal processing. Researchers interested in more elaborated algorithms can consult, for example, the techniques presented in [2].

An interesting open discussion: the labels assigned to each word, which are 0, 1, 2, and 3, in the example, can be certainly modified to improve the approximation provided by the transfer function, reducing the disturbance caused by the LSM. Which method could be employed to determine such labels in an optimal sense ? Maybe, a correlation coefficient or a linear programming technique to optimize the adjustment ?

References

[1] Lawson, C.L.; Hanson, R.J. *Solving Least Squares Problems* (Classics in Applied Mathematics). Society for Industrial Mathematics Pub. Co., 1987.
[2] Coleman, J., *Introducing Speech and Language Processing*. Cambridge: Cambridge University Press, 2005.
[3] Haykin, S. *Neural Networks and Learning Machines*. 3 ed. Prentice-Hall, 2008.
[4] Al-Akaidi, M. *Fractal speech processing*. Cambridge-UK: Cambridge University Press, 2004.
[5] Bosi, M; Goldberg, R. E. *Introduction to digital audio coding and standards*. Boston-USA: Kruwer Academic Press, 2003.