The Morphic Generator Grammatical Inference Methodology and Multilayer Perceptrons: A Hybrid Approach to Acoustic Modeling *

M. J. Castro and F. Casacuberta

Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Spain
e-mail: {mcastro, fcn}@dsic.upv.es

Abstract. A new hybrid approach to acoustic modeling based on the use of a grammatical inference technique to infer the structure of the models of the sublexical units and an artificial neural network as a means to estimate the emission probabilities of such models is presented. The chosen grammatical inference technique is the so-called "morphic generator grammatical inference" methodology and the connectionist model is the multilayer perceptron. The results on a continuous speech recognition task are better than those obtained by other systems such as semi-continuous hidden Markov models and semi-continuous stochastic grammars. The system even performs slightly better than other hybrid approaches.

1 Introduction

In syntactic pattern recognition, the interpretation of patterns as strings of primitives introduces a source of quantization errors in the identification process of the pattern [14]. In speech recognition, particularly, the set of primitives is obtained from a process of vector quantization (VQ). The space of feature vectors (typically, a space of Cepstral vectors) is divided into a finite set of clusters, and each one is represented by one codeword. An input pattern, represented as a sequence of feature vectors, is interpreted as a string of codewords (each vector is replaced by the closest codeword, using a nearest neighbor technique). Therefore, quantization errors are introduced for each vector of every pattern.

In speech recognition, discrete hidden Markov models (HMMs) are used as structural models for pattern recognition. In a first approach, some types of these models can be considered as stochastic regular grammars [2]. Continuous and semi-continuous HMMs [11, 12], which are extensions of the discrete models, have been introduced in order to use the continuous feature space directly with structural models. These models do not require the codification of the input pattern as a string of primitives. Continuous HMMs model the continuous feature vectors directly, using continuous probability density functions (PDFs). These models usually require mixtures of a large number of PDFs. Obviously, while

* Work partially supported by the Spanish CICYT under contract TIC95/0884-C04.

continuous HMMs avoid the errors derived by the VQ process, the use of continuous PDFs increases the computational costs of the corresponding algorithms. The semi-continuous HMMs are special types of continuous HMMs, where the PDFs associated to every state are tied. The mixtures are linear combinations of a set of independent PDFs. In this approach, the general assumption is that the entire parametric space of representation is covered by this set of independent PDFs, each one corresponding to a VQ-codeword distribution.

A more recent approach for dealing with the problem of the use of the continuous feature space is based on hybrid systems formed by HMMs and artificial neural networks [1]. In this case, multilayer perceptrons (MLPs) are used to estimate posterior probabilities. This proposal has the advantage that MLPs can better approach arbitrary density functions than linear combinations of a (small) number of PDFs.

However, the main drawback of the application of the HMM approach is that the structural component (states and transitions) requires heuristic tailoring by hand, using some a priori knowledge and/or experimentation. Actually, most of the usually adopted structures are very simple and the good performance of the models rely on the ability to estimate their parameters from training samples (Baum-Welch, Viterbi re-estimation, etc...). One consequence of this is that no standard structure has yet been commonly accepted, even for very simple speech recognition tasks.

In syntactic pattern recognition, and therefore in speech recognition, formal language theory offers an alternative framework to hidden Markov modeling (in fact, these are particular cases of stochastic finite state automata). This theory allows for the use of powerful techniques for learning the grammar associated to a class of patterns. These are known as grammatical inference techniques [6, 9]. Nonetheless, these techniques generally do not take into account the above mentioned problem of VQ errors and only deal with a symbolic (discrete) representation of patterns. One of these grammatical inference techniques is the "morphic generator grammatical inference" (MGGI) methodology which was introduced in [8]. The former version of this methodology used a discrete representation of patterns, that is, strings of codewords, and a very primitive error-correcting procedure.

A semi-continuous approach to the MGGI was proposed in [14] to combine a grammatical inference methodology with a technique to avoid the errors that are produced in the codification process of the input patterns. On one hand, a stochastic regular grammar was learned from a finite set of strings of codewords by using the MGGI methodology. Then, a continuous PDF substitutes the terminal symbol of each rule of the grammar in a similar, but not identical, way as in the semi-continuous HMM approach. These PDFs are obtained from the continuous information of the input training patterns. And finally, the parsing with such grammars only needs sequences of feature vectors. This semi-continuous MGGI methodology has been successfully used to infer regular grammars for some specific tasks in speech recognition [7].

Here, an alternative approach is studied in the same way as hybrid HMMs and artificial neural networks systems, but using stochastic regular grammars obtained by the MGGI methodology instead of HMMs. Comparative experiments have shown better results with this approach than the semi-continuous MGGI and the conventional semi-continuous HMM ones. Its performance is even slightly superior to a hybrid HMM and artificial neural networks system.

2 The MGGI Methodology

The MGGI methodology was originally proposed as a step towards the inference of regular grammars [8]. This methodology is directly based on the concept of *local language* and the property which relates local languages and general *regular languages*. Let us introduce the MGGI methodology defining first local languages and explaining how to infer them from samples.

Let Σ be a finite alphabet; I and F be two subsets of Σ (Initial and Final symbols, respectively); and T a subset of Σ^2 (forbidden Transitions). A *local language* associated to the four-tuple $Z = \langle \Sigma, I, F, T \rangle$, $l_2(Z)$, is a subset of $\Sigma^\star$ defined as the set of strings $x = x_1 x_2 \ldots x_{|x|} \in \Sigma^\star$ such that

$$\begin{aligned}&x_1 \in I,\\&x_{|x|} \in F, \text{ and}\\&x_i x_{i+1} \notin T, 1 \leq i < |x|;\end{aligned}$$

(i.e., the sentences of $l_2(Z)$ are characterized as beginning with an initial symbol, ending with a final symbol, and having no substring of length two belonging to the set of forbidden transitions).

Given a finite set of training samples, $R \subset \Sigma^\star$, the smallest local language containing R can be obtained by using the "local language inference" algorithm described in [8], which basically consists on:

1. Given R, associate the four-tuple $Z = \langle \Sigma, I, F, T \rangle$ as follows:

   $$\begin{aligned}&\Sigma \quad \text{is the set containing all the terminal symbols of strings of } R;\\&I = \{a \in \Sigma \mid \exists ax \in R \text{ and } x \in \Sigma^\star\};\\&F = \{b \in \Sigma \mid \exists xb \in R \text{ and } x \in \Sigma^\star\}; \text{ and}\\&T = \Sigma^2 - \{ab \in \Sigma^2 \mid \exists xaby \in R \text{ and } x, y \in \Sigma^\star\}.\end{aligned}$$

   It is easily seen that $l_2(Z)$ is the smallest local language containing R.

2. Then, given R and the four-tuple Z, build a regular grammar that generates $l_2(R)$, $\langle \Sigma, N, P, S \rangle$, where:

   $$N = \{X_a, \forall a \in \Sigma\} \cup \{S\};$$

   and the set of productions P is defined for all symbols a and b from Σ as:

   $$\begin{aligned}&S \to \lambda X_a, &&\text{iff } a \in I,\\&X_b \to b, &&\text{iff } b \in F,\\&X_a \to a X_b, &&\text{iff } ab \notin T,\end{aligned}$$

   being λ the empty string.

The main drawback of such local language inference algorithm is that it will generally lead to overgeneralized languages. The MGGI methodology is an inference scheme which attempts to avoid this undesirable property.

The MGGI methodology can be formally described as follows [8]. Let Σ and Δ be two finite alphabets, and $\Sigma^\star$ and $\Delta^\star$ are the free monoids over the respective alphabets; and R be a finite set of training samples, $R \subset \Sigma^\star$. Let g be a renaming function, $g : R \rightarrow \Delta^\star$; and h be a letter-to-letter morphism, $h : \Delta^\star \rightarrow \Sigma^\star$. The *regular language*, L, generated by the MGGI-inferred grammar G, is related to the set of training samples, R, through the expression

$$L = h(l_2(g(R))),$$

where $l_2(g(R))$ is obtained by using the above local language inference algorithm. Graphically, this methodology is illustrated in Figure 1.

$$\begin{tikzcd}
R \subset \Sigma^\star \arrow[r, "g"] \arrow[d, "\text{MGGI}"'] & g(R) \subset \Delta^\star \arrow[d, "\text{local language inference algorithm}"] \\
L \subset \Sigma^\star & l_2(g(R)) \subset \Delta^\star \arrow[l, "h"']
\end{tikzcd}$$

Fig. 1. Scheme of the "morphic generator grammatical inference" (MGGI) methodology.

The most usual renaming function g consists on adding adequate subscripts to every symbol of a given string from R. Let us propose a simple example, given a training set $R = \{aaba, abba, abbba, aabbbbaa\}$. The chosen renaming function g is defined as: the string is divided into a fixed number of intervals, say d, and then a subscript $i \in \{1, \ldots, d\}$ is added to each symbol, depending on the symbol is in the interval i. If d is fixed to 4, the result of applying g on R is

$$g(R) = \{a_1a_2b_3a_4, a_1b_2b_3a_4, a_1b_1b_2b_3a_4, a_1a_1b_2b_2b_3b_3a_4a_4\}.$$

The (incremental) application of the local language algorithm to $g(R)$ conveys to the following grammar,

$$N = \{S, X_{a_1}, X_{a_2}, X_{b_3}, X_{a_4}, X_{b_2}, X_{b_1}\};$$

and

$$\begin{array}{lll}
S \rightarrow X_{a_1}, & X_{a_1} \rightarrow a_1X_{a_2}, & X_{b_1} \rightarrow b_1X_{b_2}, \\
X_{a_4} \rightarrow a_4, & X_{a_2} \rightarrow a_2X_{b_3}, & X_{a_1} \rightarrow a_1X_{a_1}, \\
 & X_{b_3} \rightarrow b_3X_{a_4}, & X_{b_2} \rightarrow b_2X_{b_2}, \\
 & X_{a_1} \rightarrow a_1X_{b_2}, & X_{b_3} \rightarrow b_3X_{b_3}, \\
 & X_{b_2} \rightarrow b_2X_{b_3}, & X_{a_4} \rightarrow a_4X_{a_4}. \\
 & X_{a_1} \rightarrow a_1X_{b_1}, &
\end{array}$$

The morphism h is usually defined to be the inverse of g for R, that is $h(g(R)) = R$. This choice allows us to guarantee that we actually embed the inferred MGGI-languages within suitable "extreme languages". In practice, to define h as the inverse of g simply means omitting the subscript of every symbol of every renamed string.

Therefore, the final grammar G of the example is

$$N = \{S, X_{a_1}, X_{a_2}, X_{b_3}, X_{a_4}, X_{b_2}, X_{b_1}\};$$

and

$$\begin{array}{lll}
S \rightarrow X_{a_1}, & X_{a_1} \rightarrow aX_{a_2}, & X_{b_1} \rightarrow bX_{b_2}, \\
X_{a_4} \rightarrow a, & X_{a_2} \rightarrow aX_{b_3}, & X_{a_1} \rightarrow aX_{a_1}, \\
 & X_{b_3} \rightarrow bX_{a_4}, & X_{b_2} \rightarrow bX_{b_2}, \\
 & X_{a_1} \rightarrow aX_{b_2}, & X_{b_3} \rightarrow bX_{b_3}, \\
 & X_{b_2} \rightarrow bX_{b_3}, & X_{a_4} \rightarrow aX_{a_4}. \\
 & X_{a_1} \rightarrow aX_{b_1}, &
\end{array}$$

The definition of the function g allows us to specify the task dependent features which are desired for the inferred models. In this way, the function g becomes a control mechanism which prevents overgeneralization on the basis of our a priori knowledge about the task. In particular, for the task of sublexical modeling, it seems clear that a phone model needs to represent (at least) the three different intervals of a phone; i.e. left (L, on-glide phase), middle (M, central-phase) and right (R, off-glide phase). Therefore, we can define a function g which adds the "positional" index L, M or R to each codeword of each string in R, depending on its relative position in it.

The estimation of the probability distributions of the stochastic extension of the MGGI methodology is performed by using the Viterbi re-estimation algorithm [3, 7]. The initialization is based on the frequency of the use of the corresponding transitions in the derivations of the strings of $g(R)$ in $\Delta^\star$ since the local language inference algorithm yields unambiguous regular grammars. In the following section we describe how to estimate the emission probabilities associated to each state of the MGGI-inferred stochastic grammars using an MLP.

3 MLPs as Estimators of Posterior Probabilities. Hybrid System

The emission probability associated to each state of a structural model must be estimated, that is, the probability of the observed acoustic vector x given the hypothesized state q of the model, $p(x \mid q)$. Artificial neural networks can be trained to estimate probabilities that are related to these emission probabilities. In particular, an MLP can be trained to produce estimates of the posterior probability $P(q \mid x)$ (that is, the posterior probability of the state q of the structural model given the acoustic vector x), if each MLP output is associated with a specific state of the model and if it is trained as a classifier. In this case, it has been formally proved by several authors (among others, [1, 10]), that the MLP outputs are estimates of the posterior probabilities of the output classes. That is, an output value of the MLP given the input (an acoustic vector x) is an estimation of the posterior probability $P(q \mid x)$, with q being one of the classes (states) to classify. This posterior probability can be converted to emission probabilities by applying Baye's rule to the MLP outputs:

$$\frac{P(q \mid x)}{P(q)} = \frac{p(x \mid q)}{p(x)} \ .$$

The posterior probability estimates from the MLP outputs, $P(q \mid x)$, is divided by the class priors, $P(q)$. The class priors can be estimated from the relative frequencies of each class from the information produced by a forced Viterbi alignment of the training data. The scaled likelihood $p(x \mid q)/p(x)$ can be used as an emission probability in the proposed system, since, during recognition, the scaling factor $p(x)$ is a constant for all classes and will not change the classification [1].

The advantages of this approach are the discriminate training criterion and the fact that it is no longer necessary to assume a priori distribution of the data. Furthermore, if some left and right context is used in the input of the MLP, the output values of the MLP are good estimates of

$$P(c \mid X_{t-c_l}^{t+c_r}) \ ,$$

with $X_{t-c_l}^{t+c_r} = x_{t-c_l} \ldots x_t \ldots x_{t+c_r}$.

4 Experiments

4.1 Experimental Environment

The decoding experiments are performed with a Spanish continuous speech database, FRASES [4]. This database consists of 120 phonetically balanced sentences and 50 sentences obtained from current Spanish narrative. All the sentences were uttered by 10 speakers for a total of 1,700 sentences and about 50,000 phonemes. This database was acquired at 16 kHz and was parametrized obtaining 11-dimensional acoustic vectors (10 Cepstral coefficients and energy). A VQ process was followed in order to obtain a 32-sized codebook from the acoustic vectors. All the sentences were automatically transcribed into sequences of phones. The set of phones was composed by 23 units (that roughly correspond to the 24 Spanish phonemes) [7] plus 3 units to adequately model three types of silences (initial, final, and intermediate pauses).

In order to perform different experiments, the following distribution of the acoustic data was done. For training purposes, 840 utterances that corresponded to 120 phonetically balanced sentences uttered by 7 speakers (4 females and 3 males) were used. Out of this training set, a subset of phonetically-balanced utterances was randomly selected as a validation set (20% of the total training data). For testing purposes three different test sets were defined:

- Speaker-dependent and vocabulary-independent (SDVI): 350 utterances that corresponded to 50 sentences from narrative uttered by 7 speakers (out of the training speakers set).

- Speaker-independent and vocabulary-dependent (SIVD): 360 utterances that corresponded to 120 phonetically balanced sentences uttered by 3 (different) speakers (2 males and 1 female).
- Speaker-independent and vocabulary-independent (SIVI): 150 utterances that corresponded to 50 sentences from narrative uttered by 3 (different) speakers (2 males and 1 female).

A segmentation of the training data (that is, an assignation of acoustic subsequences to phones) was available. A small part (77 utterances) was manually segmented and the rest was automatically segmented using a classical discrete HMM system [15] bootstrapped with the manual segmentation. Every experiment was performed twice, under different conditions: without phonotactic language model and with a bigram matrix of phones obtained through a large Spanish text corpus [4].

4.2 Architecture and Training of the Hybrid System

The stochastic regular grammars inferred by the MGGI methodology (one grammar for each phone) were obtained from the segmented training data. Every phone model was inferred using a renaming function g which adds a positional index (Left, Middle or Right) to each codeword of every training sample, except for the silence models which were labeled using only one interval. The mean number of states of the inferred models was 65, and their average branching factor was 6.

As it is commented in Section 3, to obtain (quantities proportional to) the emission probabilities (tied within each inferred model), the posterior probability of each of the 26 phones (estimated through the MLP)[2] was divided by its prior probability [1]. This last probability was estimated as the relative frequency of the acoustic vectors corresponding to the phone, given the current segmentation.

In the training process of the MLP, the desired outputs were 1 if the original acoustic vector, according to the current segmentation, corresponded to the phone whose posterior probability must be obtained as the output, and it was 0, otherwise. The MLP input layer was formed by 99 inputs corresponding to the current acoustic vector (11 inputs) and four acoustic vectors of left and right context (scaled to the [0,1] interval), while the hidden layer consisted of 100 units. The training of the MLP was performed using the on-line scheme of the backpropagation algorithm [13] with a sigmoidal function, and the criterion function was the mean squared error. To prevent overtraining, after each epoch, the classification performance at acoustic vector level was measured on the validation set and the training process of the MLP was stopped when no improvement was expected.

[2] In order to verify stochastic constraints, a normalization over all outputs was performed.

5 Results and Concluding Remarks

The results of the experiments are reported using the "percent total" assessment parameter, $Pt = c/(c + s + i + d)$, where c is the number of correctly recognized phones, and i, s, and d are the number of insertions, substitutions, and deletions, respectively. This parameter was obtained by a dynamic programming algorithm for editing the output of the decoder and the correct phonetic transcription of each test utterance (without taking silences into account).

In Table 1, the experimental results with the hybrid MGGI-MLP system (without phonotactic language model) along with the results obtained with other methodologies are shown. The same experiments with bigrams of phone units are reported in Table 2. The results obtained with the MGGI-MLP system are better than those obtained through semi-continuous HMMs and semi-continuous MGGI systems (these experiments were carried out by adding to the each feature of the acoustic vectors its respective first derivative) [7]. Even the performance of the presented system is slightly superior than when a hybrid HMM-MLP system is used [5]. Furthermore, we expect to improve the proposed hybrid system by estimating the emission probabilities for each model more accurately, with a less restrictive tying of the posterior probabilities. This can be achieved by estimating the posterior probabilities of each state of the models taking into account not only the acoustic vector but also the codeword associated to the state.

Table 1. Recognition results (in %) of the decoding experiments of the three test sets (SDVI, SIVD, SIVI) without phonotactic language model.

Test set	HMM	MGGI	HMM-MLP	MGGI-MLP
SDVI	66	73	75	76
SIVD	65	65	67	69
SIVI	62	63	66	69

Table 2. Recognition results (in %) of the decoding experiments of the three test sets (SDVI, SIVD, SIVI) with bigrams language model.

Test set	HMM	MGGI	HMM-MLP	MGGI-MLP
SDVI	67	73	77	76
SIVD	65	66	70	71
SIVI	64	66	70	71

Acknowledgment

The authors thank to the anonymous refeers the carefully review of this article.

References

1. H. Bourlard and N. Morgan. *Connectionist speech recognition: A hybrid approach*, volume 247 of *Series in engineering and computer science.* Kluwer Academic, 1994.
2. F. Casacuberta. Some relations among stochastic finite state networks used in automatic speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):691–695, 1990.
3. F. Casacuberta. Growth transformations for probabilistic functions of stochastic grammars. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(3), 1996.
4. M. J. Castro. Condiciones experimentación sobre la base de datos fonética FRASES. Technical report DSIC II/21/95, Universidad Politécnica de Valencia, 1995.
5. M. J. Castro and F. Casacuberta. An acoustic-phonetic decoder for Spanish continuous speech recognition based on a connectionist-hidden Markov modelling. In *VI Spanish Symposium on Pattern Recognition and Image Analysis*, pages 301–307, Córdoba, Spain, 1995.
6. K. S. Fu. *Syntactic pattern recognition and applications.* Prentice Hall, 1982.
7. I. Galiano, E. Sanchis, I. Torres, and F. Casacuberta. Acoustic-phonetic decoding of Spanish continuous speech. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):155–180, 1994.
8. P. García, E. Vidal, and F. Casacuberta. Local languages, the succesor method, and a step towards a general methodology for the inference of regular grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):841–845, 1987.
9. R. C. González and M. G. Thomason. *Syntactic pattern recognition: An introduction.* Addison Wesley, Reading, MA, 1978.
10. H. Ney. On the probabilistic interpretation of neural network classifiers and discriminative training criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):107–119, 1995.
11. X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov models for speech recognition.* Edinburgh University Press, 1990.
12. L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition.* Prentice-Hall, 1993.
13. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *PDP: Computational models of cognition and perception, I*, chapter Learning internal representations by error propagation, pages 319–362. MIT Press, 1986.
14. E. Segarra, I. Galiano, and F. Casacuberta. *Advances in structural and syntactic pattern recognition*, volume 5 of *Machine perception and artificial intelligence*, chapter A semi-continuous extension of the morphic generator grammatical inference methodology, pages 184–193. World Scientific, 1992.
15. I. Torres, A. Varona, and F. Casacuberta. Automatic segmentation and phone model initialization in continuous speech recognition. *Proceedings in Artificial Intelligence*, I:286–289, 1994.