Inductive String Template-Based Learning of Spoken Language

Alexander Gutkin and Simon King

Centre for Speech Technology Research, University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW, United Kingdom
{Alexander.Gutkin, Simon.King}@ed.ac.uk

Abstract. This paper deals with formulation of alternative structural approach to the speech recognition problem. In this approach, we require both the representation and the learning algorithms defined on it to be linguistically meaningful, which allows the speech recognition system to discover the nature of the linguistic classes of speech patterns corresponding to the speech waveforms. We briefly discuss the current formalisms and propose an alternative — a phonologically inspired string-based inductive speech representation, defined within an analytical framework specifically designed to address the issues of class and object representation. We also present the results of the phoneme classification experiments conducted on the TIMIT corpus of continuous speech.

1 Introduction

One of the issues often neglected during the design of the speech recognition systems is the issue of whether the learning method can actually discover the representation of the class of patterns in question, in other words, to attempt to derive the structural make-up of the patterns which would allow to form some "idea" about the observed acoustic sequence. The representation is better to be structural simply because the use of vector spaces for modeling does not allow to go beyond construction of hyperspaces which are semantically uninformative. In Sect.2 we provide the formulation of some of the requirements for such representations based on the requirements of the inductive learning process, put forward in [1]. We require such a representation to be linguistically meaningful (interpretable) and provide means of inductive class-description (potentially being able to generate new objects belonging to the class).

In Sect. 3 we give an outline of a *rigid* structural representation, given by a pseudo-metric space, a pair consisting of a set of phonological templates plus some dissimilarity measure defined on them. We choose to base our analysis on the concept of *distinctive phonological features* — the fundamental unit of linguistic analysis. In analytical terms, this concept is necessary for fully and economically describing various phonemic properties of speech. We also assume that these features can be reliably recovered from the acoustics, the assumption supported by the recent encouraging results reported in speech recognition literature [2–4]. Finally, we view the pattern recognition models recovering this information from speech as structure detectors. The procedure of recognizing unseen patterns in the symbolic space is therefore conducted by template matching using symbolic metric algorithms [4]. It can be readily verified, that the object representation with the set of dissimilarity metrics operating on the objects thus defined, is not meaningful in artificial intelligence terms. Firstly, it does not provide us with any means of class description (normalized edit distance between the templates, for instance, does not furnish us with any understanding of the structural makeup of the particular class of phones in question). Moreover, even if we learn the optimal weights for the dissimilarity measures from the training data (thus achieving better separation between the classes), we would still not be able to learn anything about *what* makes the phones structurally different, in other words their nature. Obviously, in this context, the generativity criterion is not satisfied either.

The above limitations of the rigid representation can be eliminated by casting the problem into the Evolving Transformation Systems (ETS) formalism, as demonstrated in Sect. 4, where an inductive speech representation is outlined. The transition is accomplished by augmenting the rigid representation, described above, with the analytical machinery necessary for the representation to become inductively meaningful. The ETS formalism has been specifically developed to address the needs of an inductive learning process [5, 6]. One the central ideas of this formalism is that the similarity measure plays the critical role in the definition of a class [6] via capturing the compositional makeup of objects. We chose not follow too formal an exposition, basing the exposition on [7, 1] (more formal approach is taken in [5]). With help of this formalism we are able to discover the inductive string-based structure of various classes of phonemes.

Experiments conducted on the TIMIT corpus of continuous speech are described in Sect. 5. We conclude the paper in Sect. 6 and discuss future research work aimed at improving our representation and rectifying some of the problems with the existing approach.

2 Problem Formulation

Given a finite set C^+ of positive training objects that belong to a (possibly infinite) set C (concept) to be learned and a finite set C^- of negative training objects that do not belong to the concept C, find an analytical model that would allow one to construct the class representation and, *as a consequence*, to recognize if the new element belongs to C. In other words, on the basis of a finite training set $C^+ \cup C^-$ such that $C^+ \cap C^- = \emptyset$, where $\emptyset$ is an empty set, the agent must be able to form an "idea" of the inductive generalization corresponding to the concept C.

The *structure of a class* is taken to be:

1. The *symbolic* features that make the objects of the same class similar to each other and/or different from other objects outside the class.
2. The *emergent* combinative interrelationships among these features.

The inductive learning process would then involve the discovery and encoding of the structure of the class allowing to abstract (generalize) and associate meaning with the set of objects. In the consequent recognition stage, the *induced* dissimilarity measure is used to compare a new object to some fixed and reduced set of objects from C^+.

3 Basic Phonological Object Representation

The lowest (closest to acoustics) level of linguistic hierarchical representation of an utterance is usually represented by *phonological distinctive features* [8], which are seen in various phonological theories as the atomic units fully and economically describing the phonemic inventory of any given language. Phonemic inventory (usually consisting of a few dozen categories), in turn, is used to describe the possibly unlimited range of sounds (phones or segments) encountered in spoken language. Any *phoneme* is seen as minimal contrastive sound unit of a language (two phones are different phonemes if they produce phonological contrast), is thus represented as a bundle of simultaneous atomic units, the sum of properties of which makes a phoneme. The distinctive features used in this work are multi-valued. Each of the N features takes one of the several possible values — for example, manner of articulation is one of: approximant, fricative, nasal, stop, vowel, silence.

While at present we cannot extract structural information from the waveforms, there is a feasible alternative which appears to be a reasonable way to proceed. In approach described in [3], phonological feature recovery from speech waveforms is performed by time-delaying recurrent neural networks whose activation values are interpreted as probabilities of certain features being present in the sound corresponding to the current frame. Since each probability measurement recovered in this way has a direct linguistic interpretation, we assume that this numeric measurement corresponds to a certain linguistic fact and can thus be represented symbolically, turning the neural networks into an effective structural/logical detector. An algorithm described in [4], was used to map the continuous activation values of the neural networks into the symbols, using simple quantization over N separate finite alphabets of equal size for each of the N values separately.

Once the speech has been transformed into a sequence of vectors of symbols, it can be seen as a sequence of symbolic matrices, each identifying a phone in terms of its distinctive phonological features. A phone realization (token) p of class (type) P, $p \in P$, is thus represented as

$$\begin{array}{cccc} f_1^{t_p} & f_1^{t_p+1} & \dots & f_1^{t_p+k_p-1} \\ f_2^{t_p} & f_2^{t_p+1} & \dots & f_2^{t_p+k_p-1} \\ \dots & \dots & \dots & \dots \\ f_N^{t_p} & f_N^{t_p+1} & \dots & f_N^{t_p+k_p-1} \end{array} \qquad \rightarrow_t$$

where t_p is the start time, k_p is the duration of p in frames and N is the fixed number of distinctive phonological feature-values which henceforth will be referred to as *streams*. Each of the five features has multiple possible values and hence multiple corresponding streams.

This representation has a number of attractive features. It accounts for duration and contextual effects. Since the durations of tokens vary, even within a class, templates of various durations can be used for a given class. Aspects of co-articulation (such as assimilation, described above) can be accounted for, since the features are represented explicitly and independently. They can change value anywhere within a given template. Finally, this representation is amenable to human examination since its components have explicit linguistic interpretations.

Once the structural representation is obtained by means of quantization of neural network outputs, the next step is to define a dissimilarity measure between pairs of templates, or between a template and a token to be classified. An assumption made in [4] is that the streams are entirely independent of one another and all have equal importance. For a single token, each stream is a string of symbols from one of the corresponding alphabets.

Figure 1 shows a simple representation for the two-class problem consisting of /*p*/ and /*b*/ consonants, for each of which two realizations are available. Each template consists of three independent distinctive feature streams (over a three-symbol alphabet) from the SPE features system defined in [9]. The three symbols can be interpreted as feature being absent from the makeup of the phone (*low*), feature undergoing a transition (*mid*) and feature being present (*high*).

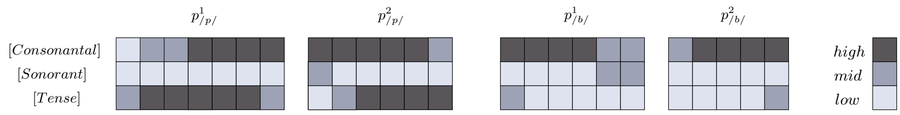

Fig. 1. Simple three-stream template representation of phones /*p*/ and /*b*/ over a three symbol alphabet

A *phonological pseudo-metric space*, corresponding to the structural representation, is a pair (P, D) where P is a set of all possible templates having N streams and $D\colon P \times P \to \mathbb{R}^+$ is a mapping of the Cartesian product $P \times P$ into the set of non-negative real numbers $\mathbb{R}^+$, such that $D = \sum_{i=1}^{N} d_i$, where d_i can be any chosen string dissimilarity measure, satisfying the conditions of reflexivity: $\forall x \in P \;\; D(x, x) = 0$ and symmetry: $\forall x, y \in P \;\; D(x, y) = D(y, x)$. The set P we consider is obviously finite. The resulting properties of the pseudo-metric space are essentially dictated by the per-stream distance functions d_i. The same type of distance function is used for all the streams.

Given an example representation in Fig. 1, and defining the weighted Levenshtein distance to act on the templates, we obtain a simple metric space where the set P consists of four templates and the metric is defined as a linear combination of three independent per-stream weighted Levenshtein edit distances over three different alphabets.

4 Template-Based Evolving Transformation System

Template Transformation System A *transformation system* (TS) is a triple $T = (P, O, D)$, where P is a set of phonological templates defined above; $O = \{o_i\}_{i=1}^m$ is a finite set of m substitution operations for transforming templates and can be thought of as a postulated set of basic, or primitive, object features, satisfying the following two conditions: all the substitution operations are reversible and for every pair of templates there exists a sequence of operations that transforms one template into the other; $D = \{\Delta_\omega\}_{\omega \in \Omega}$ is a (competing) *parametric family of distance functions* defined on P whose parameter set Ω is the $(m-1)$-dimensional unit simplex [1] in $\mathbb{R}^m$ given by

$$\Omega = \big\{\omega = (w^1, w^2, \ldots, w^m) \mid w^i \geq 0, \quad \sum_{i=1}^{m} w^i = 1\big\}$$

and each of the distance functions Δ_ω is defined as follows: weight w^i is assigned to the operation o_i and $\Delta_\omega(p_k, p_l) = \min_{o_j \in O} \sum_{i=1}^{k} w_j^i$, where the minimum is taken over the set O of all possible sequences $o_j = (o_1^j, \ldots, o_k^j)$ of operations that transform template p_k into template p_l.

For example, the templates introduced in the preceding section furnished with the weighted Levenshtein edit distance define a transformation system whose set of operations consists of single character substitutions, deletions and insertions. The set of operations is not limited to single letter operations and can include strings of length more than one known as *blocks*. In such a case, weighted Levenshtein edit distance can be extended to form a pseudo-metric called *Generalised Levenshtein Distance* [11]. In this case, the operations on phonological templates are defined in exactly the same way as in the case of a regular weighted Levenshtein distance.

The adjective "competing" is introduced to draw attention to the fact that during learning only a subset of m weights is "selected" as non-zero. That is, given a finite set of learning patterns, some weighting schemes ω are more appropriate than others for the learning class discrimination. It is the various operations O_i that actually "compete" with each other because of the condition $\sum_{i=1}^m w^i = 1$. Thus, all the properties of the system resulting from this definition can be viewed as *emergent* properties [5].

Given the sets of positive C^+ and negative C^- training templates from some finite labeled set, the learning in a transformation system reduces to optimization problem of a following weight function $f\colon \mathbb{R}^m \to \mathbb{R}$, where m is the number of operations in O,

$$\max_{\omega \in \Omega} f(\omega) \;= \max_{\omega \in \Omega} \frac{\beta(\omega)}{\epsilon + \alpha(\omega)} \;, \tag{1}$$

the function is restricted to $(m-1)$-dimensional simplex Ω given above, $\beta(\omega)$ is the Δ_ω-distance between C^+ and C^- called *interclass distance*, $\alpha(\omega)$ is the

[1] A concept from functional optimisation not to be confused with the Dantzig simplex method for linear programming [10].

average Δ_ω-distance within C^+ called average *intraclass distance* and ϵ is a small positive constant to prevent the overflow condition when the values of $\alpha(\omega)$ approach zero. Hence $f(\omega)$ combines in itself both the measure of compactness of C^+, as well as the measure of separation of C^+ from C^-, following from the simultaneous minimization of function α and maximization of function β [5, 7]. The result of the optimization process is the set of optimal vector weights $\hat{\omega} = \arg\max_{\omega \in \Omega} f(\omega)$, which generates the most distinctive metric configuration for the class within the global training set.

Evolving Metric and Inductive Class Representation When the set O of substitution operations is not sufficient to achieve a complete separation of between the classes, the structure of the model allows for the modification of the set O which is achieved by adding some new transformation operations, each representing a composition of several initial operations, obtaining new set of transformation operations and thus a new transformation system. Addition of the operations has the effect of changing the geometry of the distributions of object classes in the corresponding environment: new shorter transition paths are generated between some pairs of objects in the structured object set. This leads to the central concept of the transformation system model, the mathematical structure constructed as a sequence of transformation systems.

Evolving Transformation System (ETS) is a sequence of transformation systems, defined above, with a common set P of structured objects $T_i = (P, O_i, D_i)$ in which each set of operations O_i, except O_0, is obtained from O_{i-1} by adding to it one or several operations that are constructed from the operations in O_{i-1} with the help of a small fixed set R of *composition rules, or operators.* Each rule $r \in R$ specifies how to (systematically) construct the corresponding new operation from its operands [5, 6].

In Levenshtein string transformation system, for example, new operations can be constructed by concatenating the two left-hand sides of the given single-letter operations, yielding new string operations, i.e. given $a \leftrightarrow \epsilon$ and $b \leftrightarrow \epsilon$, the new operation is $ab \leftrightarrow \epsilon$.

From the above definition it follows that at the stage t of the learning process within the evolving transformation system, $O_0 \subseteq O_1 \subseteq \ldots \subseteq O_t$ and for $i \in [0, t-1]$

$$\forall p_k, p_l \in P,\ \forall \Delta_{\omega_1} \in D_i\ \exists \Delta_{\omega_2} \in D_{i+1} \colon \Delta_{\omega_1}(p_k, p_l) \le \Delta_{\omega_2}(p_k, p_l)\ ,$$

where $\omega_1 \in \Omega_i$, $\omega_2 \in \Omega_{i+1}$ and the dimensions of the simplex Ω_i are smaller than the dimensions of simplex Ω_{i+1}, simplex Ω_i being a sub-simplex of Ω_{i+1}, i.e. $\Omega_0 \subset \Omega_1 \subset \ldots \subset \Omega_t$. Each stage i, therefore induces a new topology represented by Ω_i.

The optimization process described in the previous section becomes an inner loop within the general inductive learning process (we are not giving details of the learning algorithm which is a variant of grammatical inference algorithm described in [7] and [11]) which proceeds by constructing a sequence of transformations $\{O_i\}$ in such a way that, for each sequentially obtained transformation system $T_i = (P, O_i, D_i)$, the inter-distances in C^+ expressed shrink to zero while the corresponding distance between C^+ and C^- remains non-zero [7]. In view of the above, following requirements of the inductive generalization are set forth in [7, 1]: The *inductive class representation* is defined as a triple $\Pi = (\hat{C}^+, \hat{O}, \hat{\Omega})$ where $\hat{C}^+ \subset C^+$, $\hat{O}$ is the final set of operations at the end of the learning process, and $\hat{\Omega} \subseteq \Omega$ is a set of optimal weight vectors $\{\hat{\omega}\}$ for the final transformation system. The elements of $\hat{C}^+$ act as reference patterns for definining the class. During classification stage, a new input pattern is always compared with these reference patterns using the set of weights $\{\Delta_{\hat{\omega}}\}$ from $\hat{\Omega}$. The set of transformations $\hat{O}$ is necessary since the concept of a distance can properly be defined only in terms of these operations.

Figure 2 shows the non-trivial stream-specific transformations discovered during the learning process for the two-class phone problem of Fig. 1. These operations (the corresponding optimal sets of weights $\hat{\Omega}_{/p/}$ and $\hat{\Omega}_{/b/}$ are not shown) together with the trivial one-symbol transformations form the optimal set of transformations for each class. Together with the corresponding sets of reference objects $\hat{C}^+_{/p/}$ and $\hat{C}^+_{/b/}$ (which for this problem consist of one template arbitrarily chosen from the corresponding training set), the three-tuples

$$\Pi_{/p/} = \left(\hat{C}^+_{/p/}, \hat{O}_{/p/}, \hat{\Omega}_{/p/}\right) \quad \text{and} \quad \Pi_{/b/} = \left(\hat{C}^+_{/b/}, \hat{O}_{/b/}, \hat{\Omega}_{/b/}\right)$$

provide inductive class representations for the two classes in question.

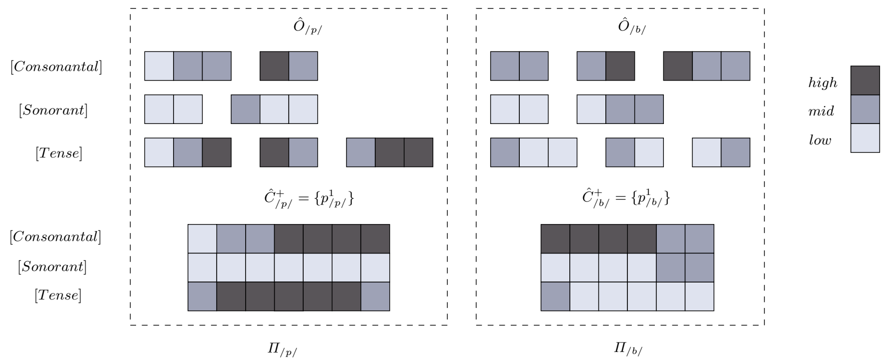

Fig. 2. Discovered per-stream feature transformations ($\hat{O}_{/p/}$ and $\hat{O}_{/b/}$) corresponding to the representation in Fig. 1 and the resulting class representations $\Pi_{/p/}$ and $\Pi_{/b/}$.

This representation is meaningful, in a sense, that it is able to capture certain consonantal properties of the phones corresponding to /*p*/ and /*b*/. From this toy example we, for example, can learn the main difference (within the postulated three-stream representation) between the two classes, namely different behavior of the [*tense*] feature. In general, tense sounds are produced with a deliberate, accurate, maximally distinct gesture that involves considerable muscular effort; non-tense sounds are produced rapidly and somewhat indistinctly. In

Fig.2, transformations corresponding to the [*tense*] stream capture the fact that within the available prototypes of /*p*/, this feature is either *high* or in the process of gradually changing around the *high* values, whereas for the prototypes of /*b*/, the process is opposite. This coincides with the assumption of phonological contrast between /*p*/ and /*b*/ phones within the SPE feature system [9]. In addition, the transformations capture certain asynchronies in the process of sound changes. The first transformation corresponding to the [*consonantal*] stream for the class /*p*/ indicates the change from *low* to *high* which most probably means that one (or both) of the prototypes were derived from the context in which they were preceded by a vowel or consonantal sounds from /*w*/ or /*j*/ classes.

5 Experiments

Our experiments used the TIMIT database [12]. This is a corpus of high-quality recordings of read continuous speech from North American speakers. The entire corpus is reliably transcribed at the word and surface phonetic levels. For details of the feature-detecting neural networks, please refer to [3]. The standard training/test data partition is kept, with only the `sx` and `si` sentences being used, resulting in 3696 training utterances from 462 different speakers, out of which 100 sentences were held out for cross-validation training of neural networks. The entire test set of 1344 utterances from 168 speakers was used for the classification experiment. None of the test speakers are in the training set, and hence all the experiments are open and speaker independent. There are 39 phone classes.

We quantised the neural network output activations using the quantisation level of 10 and removed the redundant tokens from training and test sets. The sizes of the symbolic training and test sets thus obtained are 124962 and 46633 tokens, respectively. In order to obtain the sets C^+, each training set P was reduced to 5 cluster centroids using k-medians clustering employing the Levenshtein weighted edit distance for similarity computations and set median algorithm for template selection. The clustering algorithm initialisation criteria was duration-based [4].

During the learning stage, for each class P out of the 39 classes, represented by its training set C_P^+, we derived its corresponding inductive structure Π_P by using an algorithm outlined in Sect. 4. We defined the stopping criterion for the optimisation problem to be $\lambda = f^{-1}(\hat{\omega})$, where $f(\hat{\omega})$ is given by (1). The particular value of λ we used was 10^{-8}. During the recognition stage, an efficient k-NN AESA search technique [13] was used to compare each of the 46633 test tokens with the class prototypes by using the template-based Generalised Levenshtein Distance defined by the respective class inductive structure. The classification accuracy we obtained was 51%, correctly classifying 23783 out of 46633 tokens.

6 Conclusions and Future Work

In this paper we gave an outline of a linguistically inspired structural representation for speech, an attempt to find an inductively meaningful definition for the speech recognition problem, focusing on a low level phonological representation of speech patterns. We showed how inductively "rigid" representation can be made expressive with the introduction of *evolving* metric and described the results of the initial experiments conducted with the highly non-trivial continous speech data. We believe that the emphasis on the class representation of linguistic phenomena will facilitate the development of the speech recognition field, since the recognition problem cannot be approached adequately without a meaningful representation.

There are several ways of improving the representation described in this paper. For example, instead of using extensions of standard string-based dissimilarity measures, such as weighted Levenshtein distance, we can introduce linguistically inspired distance functions along the lines of [14]. The learning algorithms, developed in the grammatical inference setting [7, 11], can be further improved to take into the account the stream-based phonological structure. In addition, some basic phonological constraints can potentially be introduced (stream independence assumption, for instance, can be relaxed to account for similar classes of distinctive phonological features, place of articulation being one of them). An efficient prototype selection algorithm for reducing the training set and selecting the templates containing inductively "interesting" features is also needed. It is expected that the above modifications will lead to significant improvements in the classification accuracy on the TIMIT task.

Acknowledgments

The authors would like to thank Lev Goldfarb and Mirjam Wester for many useful suggestions.

References

1. Goldfarb, L., Deshpande, S.S., Bhavsar, C.: Inductive Theory of Vision. Technical Report TR96-108, Faculty of Computer Science, University of New Brunswick, Canada (1996)
2. King, S., Taylor, P.: Detecting phonological features in continuous speech using neural networks. Computer Speech and Language **14** (2000) 333–353
3. Wester, M.: Syllable classification using articulatory acoustic features. In: Proc. Eurospeech, Geneva (2003) 233–236
4. Gutkin, A., King, S.: Structural Representation of Speech for Phonetic Classification. In: Proc. 17th ICPR. Volume 3., Cambridge, UK (2004) 438–441
5. Goldfarb, L.: On the foundations of intelligent processes – I. An evolving model for pattern learning. Pattern Recognition **23** (1990) 595–616
6. Goldfarb, L.: What is distance and why do we need the metric model for pattern learning ? Pattern Recognition **25** (1992) 431–438
7. Goldfarb, L., Nigam, S.: The Unified Learning Paradigm: A Foundation for AI. In Honavar, V., Uhr, L., eds.: Artificial Intelligence and Neural Networks: Steps toward Principled Integration. Academic Press, Boston (1994) 533–559

8. Jakobson, R., Fant, G.M., Halle, M.: Preliminaries to Speech Analysis: The distinctive features and their correlates. MIT Press, Cambridge, MA (1963)
9. Chomsky, N., Halle, M.: The Sound Pattern of English. MIT Press, Cambridge, MA (1968)
10. Lange, L.H.: 10. In: Elementary Linear Algebra. John Wiley & Sons, New York (1968)
11. Abela, J.M.: ETS Learning of Kernel Languages. PhD thesis, Faculty of Computer Science, University of New Brunswick, Canada (2001)
12. Garofolo, J.S.: Getting Started with the DARPA TIMIT CD-ROM: an Acoustic Phonetic Continuous Speech Database. National Institute of Standards and Technology (NIST), Gaithersburgh, Maryland. (1988)
13. Juan, A., Vidal, E.: On the Use of Normalized Edit Distances and an Efficient k-NN Search Technique (k-AESA) for Fast and Accurate String Classification. In: Proc. 15th ICPR. Volume 2. (2000) 680–683
14. Kondrak, G.: A New Algorithm for the Alignment of Phonetic Sequences. In: Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Seattle (2000) 288–295