TOWARDS CONTINUOUS ONLINE LEARNING BASED COGNITIVE SPEECH PROCESSING

Konstantin Markov

Human Interface Lab.,
University of Aizu,
Fukushima, Japan

ABSTRACT

Despite the substantial progress of the speech processing technology, it is generally acknowledged that we have a long way to go before developing ASR systems which exhibit performance approaching that of humans. Many researchers believe that simply extending our current theories and practical solutions may never lead us to that goal. One promising research direction is development of learning algorithms exhibiting human-like learning behavior. There is an apparent discrepancy between the way humans acquire their language and the way we train our systems. Humans are "learning machines" while our current systems are actually "learned machines". The ability to learn and reason in a continuing loop is attributed to the emerging cognitive systems. In this paper, we present our approach and ideas for future research in developing a cognitive speech processing system. This system has a hierarchical structure where each layer works according to the same algorithm, but represents different space or level of abstraction. The lowest layer corresponds to the acoustic space and the other layers - to phonetic, word and phrase spaces respectively. The information between layers flows in both directions: bottom-up during recognition and top-down during generation, i.e. synthesis. Development of the full system poses multiple research challenges and problems discussed in this paper.

Index Terms— Online learning, Never-Ending learning, Learning machines, Speech processing.

1. INTRODUCTION

Despite the tremendous progress in spoken language science and technology humans are still far more superior in their abilities to process, recognize and understand speech. Many researchers within ASR community agree that the performance improvement observed in the recent years can be attributed to a large extend to the increase in computing power and the availability of more speech data to train the ASR systems [1]. Using more and more data, however, does not guarantee steady improvement. In fact, systems performance is asymtoting to a level far below the human level. This suggests that it may never be possible to collect enough data to fully characterize the relationship between the linguistic message and its acoustic realization, and that simply extending our current theories and practical solutions may never lead us to the desired state of affairs [2]. What is needed is a change of the approach or even the paradigm.

Learning is an intrinsic human capability. We learn all the time, processing huge amounts of mostly unlabeled multimodal data coming from uncontrolled environments and exhibiting unlimited variability. Yet, we build our internal models of speech and language effortlessly and use them not only to recognize and predict sounds and words but to improve the subsequent learning as well. How do we do all this? There is still neither complete understanding nor computational models of human learning processes. [1].

On the other hand, automatic speech recognition has been largely regarded as a classification task, where spoken input is transformed into a sequence of pre-defined classes such as words [3]. The lifespan of an ASR system typically goes through a training phase using heavily annotated data, followed by a testing or deployment phase. The system structure is usually decided manually in advance and only the number of parameters and their values are estimated during training. After that, for the most part, ASR systems do not undergo any changes. They don't acquire new knowledge from the test data. In contrast, in humans, learning and the usage of the learned knowledge are continuous and intertwined. From this point of view we can say that humans are "learning" machines, while speech systems are, in fact, "learned" machines.

This difference has been noticed by researchers long ago [4], [5], and although it didn't attract active mainstream research, in many studies unsupervised continuous learning approaches have been investigated, especially in the neural networks community [6],[7], [8]. The main reason for the sluggish interest in this area is probably the fact that the traditional "train/test" engineering paradigm has been quite successful in wide variety of research tasks.

Inspired by the cognitive systems idea and following the results of some studies on self-organizing neural networks [9],[10][7],[11], we developed a network of hidden Markov

states called Dynamic Hidden Markov network (DHMnet) [12]. It is capable of on-line unsupervised adaptive learning and preserving previously acquired knowledge. It has dynamic structure which can grow or shrink according to the changes in the input patterns distribution, i.e. it can model non-stationary time-varying probability density functions. In this paper, we propose a system structure which is based on the DHMnet and aimed at recognizing and synthesizing speech utterances. It has several hierarchical layers, each modeled by a DHMnet, representing different space or level of abstraction. The lowest layer corresponds to the acoustic space and the other layers - to phonetic, word and phrase spaces respectively. The information between layers flows in both directions: bottom-up during recognition and top-down during generation, i.e. synthesis. The system is not yet fully implemented because there are many open questions especially on the word and phrase spaces. Nevertheless, since the information processing algorithms are quite consistent throughout all the layers, solutions developed for the first and second layer can be easily applied in the upper layers design.

2. CURRENT LEARNING PARADIGM

The speech recognition task is widely cast as a statistical classification problem where the recognition result $\hat{Y}$ is found from

$$\hat{Y} = \max_{Y} P(Y|X) = \max_{Y} P(X|Y)P(Y). \qquad (1)$$

Here, X is a representation of the input speech signal and Y is a sequence of words, phonemes, or speakers, depending on the task. There are some assumptions which, in fact, allow as to use the current train/test development paradigm. These assumptions are:

1. Random processes we deal with are stationary. This means that these processes can be described by probability density functions (pdf) with fixed structure and constant parameters. Consequently, the models we build for those pdf would have fixed structure and constant parameters as well.

2. Random samples are independently and identically distributed (i.i.d.). This is very important assumption because it justifies the whole train/test procedure. Test samples are drawn from the same distribution as the train ones and since all the samples are i.i.d., they don't need to be collected at the same time. That is, we can first train our system and then test it.

3. Classes are known in advance and their number is fixed. We may or may not have prior knowledge about the task at hand, but the minimum required is the number of classes. Furthermore, we take it as granted that this number don't change. This ensures that every test sample belongs to one of the classes.

In practice, however, none of these assumptions is true. Processes are non-stationary or at best slowly varying. Consequently, samples are not i.i.d. and if collected at different occasions may have quite different distribution. Classes are, generally, unknown and even if some prior knowledge about them is available, their number may changes.

As many researchers agree, the only viable solution to this problem is adaptation. It has to be noted, however, that all popular adaptation algorithms, such as MAP, MLLR and others, are still based on the above assumptions and therefore after the adaptation, we end up again with models which have fixed structure and constant parameters as after the initial training. When test environment changes again, we face the same problem.

3. CONTINUOUS ONLINE LEARNING

One way to approach the problem described in the previous section is to take more realistic assumptions. First we can assume that random processes are non-stationary, i.e. their joint probability distribution changes in time and space. This means that data samples are not i.i.d. anymore. In addition, we need to accept that the number of classes may be unknown and varying.

Under the new assumptions and in correspondence with Eq. (1), we can now define the classification task as

$$\hat{Y}_t = \max_{Y_t} P_t(Y_t|X_t), \qquad (2)$$

where we have assumed that:

1. Probability distributions are time-varying.

2. Samples drawn from time-varying distribution are not i.i.d.

3. Number of classes is unknown and changing in time.

For clarity, we are going to skip the time index of X_t and Y_t and focus on $P_t()$. There is no established and well understood way of modeling time-varying probability distributions yet. One approach is to assume that at a very short time scale the process is stationary, treat $P_t()$ as constant on a small time interval and try to apply classical modeling methods [13]. Another way, which is adopted in this paper, is to develop a model with time-varying structure and changing parameters. The learning algorithm of this model should then constantly track changes in the distribution as they happen. It should be able to detect new events or classes and learn them. Learning of new information, however, should not completely destroy previously learned knowledge as the adaptation does for example. Of course, some kind of knowledge erasure is unavoidable and even desirable. Apparently, learning has to be performed on-line - a case where input samples have no labels or at most are roughly annotated. This suggests an unsupervised or semi-supervised learning. We can then summarize all these requirements as follows: the model of $P_t()$ should be capable of:

1. On-line un-/semi-supervised adaptive learning.

2. Novelty detection.

3. Knowledge preservation.

4. Gradual forgetting.

Such continuous online learning is sometimes called *Never-Ending or Life-Long* learning. The above requirements give rise to the so-called *stability-plasticity* dilemma [6] - How can a system preserve its previously learned knowledge while continuing to learn new things? Several solutions to this problem have been proposed in the neural networks research field, including Adaptive Resonance Theory (ART) [4], Life-long Learning Cell Structures [7] and Self-Organizing Incremental Neural Network [11]. Commonly, network plasticity is ensured by adding new nodes to accommodate the new knowledge, while decreasing learning rates for the connection weights provides the necessary network stability. Unfortunately, such neural networks do not work with spatio-temporal data such as speech patterns.

Previously, we developed the so called Dynamic Hidden Markov network (DHMnet) [12], [14] where we tried to implement the never-ending learning requirements and avoid the limitations of the existing approaches. It is intended to be the basic building block of a new cognitive speech processing system.

4. DYNAMIC HIDDEN MARKOV NETWORK

4.1. General structure

The DHMnet consists of hidden Markov states with self-loops and transitions between them. Additionally, neighboring states are connected with lateral connections. Each state represents a part of the input feature space modeled by a multivariate Gaussian function. State sequences or paths through the network correspond to learned speech patterns or classes of patterns. Similarly to other approaches, network plasticity is ensured by adding new states and transitions whenever a new pattern is encountered. The practical problem is to define what should be considered as a "new" pattern and how to detect it. Inevitably, spurious events and noises would allocate states that may never be visited again. Such states (and paths) are considered "dead" and will be gradually removed from the network. The schematic structure of the DHMnet is shown in Fig.1, where transitions of a learned path are represented by directed solid lines, new paths with directed short dashed lines, and "dead" paths with directed long dashed lines. Undirected dashed lines represent lateral connections between states.

4.2. "Novelty" detection

Generally, any pattern that is sufficiently different from those that have been already learned can be considered a new pattern. In a manner similar to other reported solutions [8],[11], we apply a threshold to the likelihood function for "novelty" detection. Since the DHMnet is a first-order Markov chain where input vectors are presumed conditionally independent,

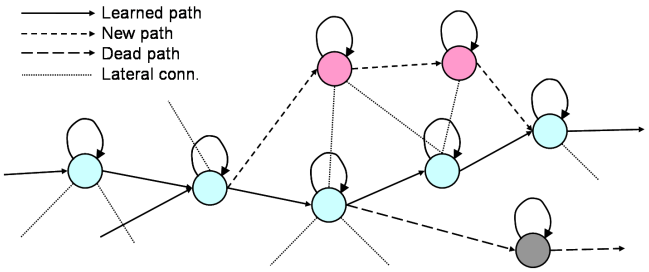

Fig. 1. Schematic structure of a Dynamic Hidden Markov network.

the pattern-level novelty detection can be substituted by multiple frame-level novelty detections. Thus, any given input vector x will be considered "new" if $P(x|\mu_b) < \theta$, where μ_b is the mean of the best matching state and the θ is the so-called *vigilance* threshold.

4.3. Stable learning

For the types of neural networks that we discussed in Section 3, the weights' update ΔW_n at each learning iteration is generally set to:

$$\Delta W_n = \alpha_n (X_n - W_{n-1}) \tag{3}$$

where X_n is the input vector and α_n is the learning rate at the n^{th} iteration. Stable learning is ensured when α_n is subject to the following constraints [11]:

$$\sum_{n=1}^{\infty} \alpha_n = \infty, \ \sum_{n=1}^{\infty} \alpha_n^2 < \infty \tag{4}$$

For the DHMnet state PDF learning, we consider the sequential version of the Maximum Likelihood estimation algorithm. In this case, the Gaussian mean update $\Delta\mu_n$ after input vector x_n will be:

$$\Delta\mu_n = \frac{1}{n}(x_n - \mu_{n-1}) \tag{5}$$

which is exactly the same as Eq.(3). The learning rate is $\alpha_n = 1/n$ and it obviously satisfies the constraints of Eq.(4).

4.4. Removing "dead" states

When a network dynamically changes its structure, the state neighborhood relations also change. To account for these changes, each lateral connection is given an age that is set to zero when a connection is made or refreshed. Otherwise, the connection age is increased every time one of the connection's states is visited. This way, connections that reach a certain age, i.e. ones that have not been refreshed for some time, are removed. The DHMnet states can have many lateral connections and if for some state all connections are removed, this state is pronounced "dead" and is removed along with all transitions to and from it.

4.5. The DHMnet algorithm

We summarize the complete DHMnet algorithm as follows:

(1) Start with an empty network.

(2) For the next input vector x_t, given the current state s_{curr}, find the best matching succeeding state s_c. If it passes the vigilance test, set it as the next state, i.e. $s_{next} = s_c$, and go to (5).

(3) Find the best state, s_a, from all other states. If it passes the vigilance test, $s_{next} = s_a$, and go to (5).

(4) Add a new state, s_t, i.e. $s_{next} = s_t$, and set its mean to x_t.

(5) Make (update) the transition from the current state s_{curr} to s_{next}.

(6) Update the means of s_{next} and all its neighbors (Eq.5).

(7) Make (refresh) the connection between s_{next} and the second best state. Increase the ages of all s_{next} connections.

(8) If any connection age has reached the age threshold, remove this connection. Remove states with no connections.

(9) Add s_{next} to the best state sequence. Set the current state $s_{curr} = s_{next}$, and go to (2).

4.6. Experiments with the DHMnet

Our preliminary experiments with the DHMnet have been already published in [12] where more details about the task and evaluation conditions can be found. Results showed that the DHMnet corresponds to the requirements for a never-ending learning system set in Section 3.

5. COGNITIVE SYSTEM STRUCTURE

Although, the DHMnet described in the previous section is capable of modeling time-varying conditional probability distributions, the speech recognition task defined as in Eq.(2) is too complex to be successfully solved by a single network. Current ASR systems have hierarchical structure consisting of several layers each modeling speech at different time scales. We take similar approach in building speech processing system based on the DHMnet. Formally, we start from Eq.(2) and factorize $P_t(Y|X)$ using several latent variables V, F and S

$$\begin{aligned} \hat{Y} &= \max_Y P_t(Y|X) \qquad (6) \\ &= \max_Y \sum_S \sum_F \sum_V P_t(Y|V)P_t(V|F)P_t(F|S)P_t(S|X) \\ &\approx \max_{Y,V,F,S} P_t(Y|V)P_t(V|F)P_t(F|S)P_t(S|X). \end{aligned}$$

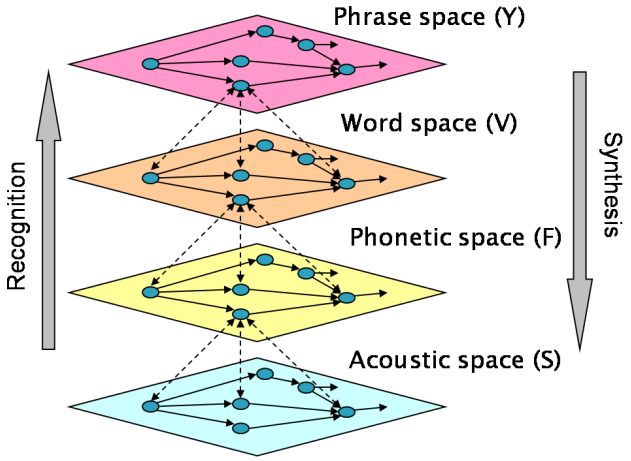

Fig. 2. Hierarchical structure of the cognitive speech processing system consisting of several Dynamic Hidden Markov networks.

Here, all variables represent sequences of variable length. The number of factor pdfs may be different, but in this case we have selected four each of which can be modeled by a separate DHMnet. They form a chain where the output of one network serves as an input to the next. For example, the output of the DHMnet representing $P_t(S|X)$ is the best state sequence S given the input X. It in turn is the input of the next DHMnet representing $P_t(F|S)$ whose output F is also input for the next network and so on. This can be viewed as an hierarchical structure consisting of several DHMnets as shown in Fig. 2. Let's for a moment assume that each of these networks represents the acoustic, phonetic, word and phrase spaces as depicted in the figure. Then, states of the phonetic space DHMnet will represent phoneme-like units. States of the word space DHMnet will have meaning of words or parts of words and the states of the phrase network will represent phrases of different length.

Following the input-output relations between DHMnets we can see that a sequence of states in the acoustic space is a point in the phonetic space and will be represented by the nearest phonetic state. In turn, sequence of phonetic states is a point in the word space and will belong to the nearest word state. Further up in this hierarchy, word state sequences will form points in the phrase space and be represented by the corresponding phrase state. Finally, phrase state sequences interpreted as the linguistic message conveyed by the acoustic signal form the output of the system. In short, any input signal of sufficient length will at the end activate one or several sequential phrase states which corresponds to *recognition* operation performed by the whole system. Information processing in this case goes from bottom-up.

By reversing the direction of the information flow, we can perform speech *synthesis* as well. Indeed, sampling from the probability distribution associated with a particular phrase

state, we get a sequence of word states. Going down one level, for each word state in the sequence we sample from its pdf to obtain sequences of phonetic states from which in turn we get sequences of acoustic states. Generating speech waves from sequences of HMM states is the main technique of the HMM-based speech synthesis technology [15]. Thus, using the same system we can perform both speech recognition and speech synthesis tasks.

So far we have assumed that each DHMnet is learned to represent particular space - acoustic, phonetic, etc. Obviously, if from the very beginning, when the system hasn't acquired any knowledge at all, we leave it to learn in a totally unsupervised manner, there is no guarantee that after some time spaces learned by each DHMnet will represent what they are meant to represent. Apparently, a more intelligent method of spoken language acquisition through unsupervised or lightly supervised manual intervention is necessary to build system knowledge. The more knowledge is acquired by the system the more capable it will become in dealing with unannotated input, something called *self-teaching* or *self-learning* [16]. There has been some research in unsupervised acquisition of words [17], [18], but there remain many open questions to be investigated in pattern discovery and learning algorithms.

Another issues of practical value concern the form of the state probability functions of the phonetic, word and phrase DHMnets. They should be such as to allow on-line parameter updates and be computationally efficient. Closely related problem is the definition of the distance measure for each space. At the lower, acoustic space, we have experimented with the standard Euclidean distance, but the solution is not that simple for the other spaces. While phoneme like units can still be compared "acoustically" by obtaining DTW score between the corresponding sequences of acoustic states means, efficient word and especially phrase comparison will require intensive further research.

6. CONCLUSION

In this paper, we presented brief analysis of the current ASR systems learning paradigm and argued that at the bottom of many practical problems we face are several assumptions tipical to the traditional classification task. We described our ideas and first steps towards development of an intelligent cognitive speech processing system. There are still a lot of questions which remain to be answered and a lot of work is necessary to even prove the feasibility of our approach.

7. REFERENCES

[1] Odette Scharenborg, "Reaching over the gap: A review of efforts to link human and automatic speech recognition research," *Speech Communication*, vol. 49, pp. 336–247, 2007.

[2] Roger Moore, "Spoken language processing: Piecing together the puzzle," *Speech Communication*, vol. 49, pp. 418–435, 2007.

[3] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of IEEE*, vol. 64, no. 4, pp. 532–557, 1976.

[4] G. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *Computer*, pp. 77–88, Mar. 1988.

[5] Dominique Beroule, "The never-ending learning," in *Neural Computers*, vol. F41 of *NATO ASI Series*, pp. 219–230. Springer-Verlag, 1988.

[6] S. Grossberg, "Nonlinear neural networks: principles, mechanisms, and architectures," *Neural Networks*, vol. 1, pp. 17–61, 1988.

[7] F. Hamker, "Life-long learning Cell Structures - continuously learning without catastrophic interference," *Neural Networks*, vol. 14, pp. 551–573, 2001.

[8] N. Srinivasa and N. Ahuja, "A topological and temporal correlator network for spatiotemporal pattern learning, recognition and recall," *IEEE Trans. Neural Networks*, vol. 10, no. 2, pp. 356–371, Mar. 1999.

[9] S. Amari, "Field theory of self-organizing neural nets," *IEEE Trans. Syst. Man Cyber.*, vol. SMC-13, no. 5, pp. 741–748, 1983.

[10] T. Kohonen, *Self-Organizaion and Associative Memory*, Springer-Verlag, Berlin, Germany, 3rd edition, 1989.

[11] S. Furao and O. Hasegawa, "An incremental network for online unsupervised classification and topology learning," *Neural Networks*, vol. 19, pp. 90–106, 2006.

[12] K. Markov and S. Nakamura, "Never-Ending Learning with Dynamic Hidden Markov Network," in *Proc. Interspeech*, 2007, pp. 1437–1440.

[13] H. Takizawa and H. Kobayashi, "Partial distortion entropy maximization for online data clustering," *Neural Networks*, vol. 20, pp. 819–831, 2007.

[14] K. Markov and S. Nakamura, "Language Identification with Dynamic Hidden Markov Network," in *Proc. ICASSP*, 2008, pp. 4233–4236.

[15] K. Tokuda, T. Yoshimura, T. Masuko, Kobayashi T., and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Proc. of ICASSP*, 2000, pp. 1315–1318.

[16] J.M. Baker, Li Deng, S. Khudanpur, Chin-Hui Lee, J.R. Glass, H. Morgan, and D. O'Shaughnessy, "Updated minds report on speech recognition and understanding, part 2," *Signal Processing Magazine, IEEE*, vol. 26, no. 4, pp. 78–85, July 2009.

[17] A. Venkataraman, "A statistical model for word discovery in transcribed speech," *Comput. Linguist.*, vol. 27, no. 3, pp. 353–372, 2001.

[18] Alex Park, *Unsupervised pattern discovery in speech: Applications to word aquisition and speaker segmentation*, Ph.D. thesis, MIT, Cambridge, MA, 2006.