A noise robust method for pattern discovery in quantized time series: the concept matrix approach

Okko Johannes Räsänen[1], Unto Kalervo Laine[1], and Toomas Altosaar[1]

Department of Signal Processing and Acoustics, Helsinki University of Technology, Finland
Okko.Rasanen@tkk.fi, Unto.Laine@tkk.fi, Toomas.Altosaar@tkk.fi

Abstract

An efficient method for pattern discovery from discrete time series is introduced in this paper. The method utilizes two parallel streams of data, a discrete unit time-series and a set of labeled events, From these inputs it builds associative models between systematically co-occurring structures existing in both streams. The models are based on transitional probabilities of events at several different time scales. Learning and recognition processes are incremental, making the approach suitable for on-line learning tasks. The capabilities of the algorithm are demonstrated in a continuous speech recognition task operating in varying noise levels.

Index Terms: speech recognition, pattern discovery, time series analysis

1. Introduction

Current state-of-the-art approaches in automatic speech recognition (ASR) are based on Hidden-Markov Models (HMM; [1]). Although the performance of HMM based recognizers is very good in many applications, they require large amounts of training with annotated speech material. Also, mismatches between training data and the actual signal and speaker conditions during recognition impose serious problems, making the recognizers fall far behind humans, e.g., in terms of noise robustness. These are central reasons why a major body of ASR research is focused on improving existing HMM algorithms for higher noise robustness, faster learning, speaker adaptation, etc.

Contemporarily, new methods and architectures have been studied in order to complement and challenge the prevailing HMM approaches in different types of speech recognition tasks (e.g., artificial neural networks, [2], or Non-Negative Matrix Factorization, [3-5]). Additionally, it has been suggested that systems capable of self-driven structure discovery may be required for more human-like performance in many speech recognition and artificial intelligence tasks (see, e.g., [6]). By *discovering* and *memorizing* associations between internal states of a system and multimodal external input streams, in a process called *grounding* [7], the system can form information structures that can be referred to as *meanings*. When a familiar pattern is perceived the associative links, and thereby the meaning, becomes activated: the input is *recognized*. This viewpoint is different from traditional HMM-based approaches where *models* of pre-defined units are trained from data in a process that uses expert knowledge in natural language theory, engineering, and phonetics.

In this paper a novel approach for associative pattern discovery in time series is introduced. This method, called the *concept matrix* (CM) approach, combines information from two input streams in order to find co-occurrence relations between them. It *learns* recurring structures in similar contexts, and *recognizes* them from new input. Contrary to HMM, CM does not make the Markov property assumption regarding independence of the subsequent states, making it capable of finding structures between non-adjacent events and robust against temporally local distortions. We demonstrate the capabilities of this method in a weakly supervised word learning and recognition task using continuous speech. However, it should be noted that the algorithm is not theoretically limited to speech recognition and can be utilized for any kind of pattern discovery from time-series that can be expressed as discrete sequences (e.g., image recognition or medical signal processing).

2. The concept matrix algorithm

2.1 Inputs

Input to the system consists of a time series of discrete elements or sampled spatial information to form 1D-sequences, and in the training phase, tags specifying some events associated with the sequences.

The first information source consists of time-series of basic elements called *labels*. In the simplest case they may refer to items in a vector quantization (VQ) codebook, or they can be produced by discretization of time-series or images. In a more complex case they may refer to some higher-level representation of information, e.g., events or items possibly reflecting clear qualitative properties.

The other information source is represented by a set of so-called *concept tags c*. Tags are integer values that represent invariant outputs of another process that are concurrently activated with the time-series input (e.g., a categorization process performed in another modality like visual or haptic perception in case of speech recognition, or some other group of manually defined events that should be associated with the time-series; see also [8]).

The mechanism may work also in the opposite direction; an acoustic event may serve as a tag to learn visual patterns. One modality may form tags to other modalities to help learning. More generally, the method allows construction of statistical associations between different modalities. This is one of the key issues regarding the modeling and understanding of the formation and learning of *meanings* (by agents and humans).

2.2 Training

When a set of tags $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$ and a label sequence $\boldsymbol{s}$ is represented, the algorithm starts to collect frequency data regarding the occurrences of label pairs in the sequence at distances $\boldsymbol{l} = \{l_1, l_2, l_3, \ldots, l_n\}$. This data is stored into histogram tables $\mathbf{T}_{l,c}$ specified by the lag l and $\mathbf{c}$, i.e., a separate table exists for each tag at each lag, yielding a total of $N_l * N_c$ tables where N_c is the total number of all possible tags, N_l is the number of used lags. The original labels can be used as pointers to $\mathbf{T}$ when the number of occurrences of the corresponding label pair is required. This first step shares similar properties with the HAC-model used in [3].

After frequency data from $\boldsymbol{s}$ are collected, data from every $\mathbf{T}_{l,c}$ are normalized to an activation matrix $\mathbf{P}_{\mathbf{l,c}}$ of size N_q x N_q, where N_q is the size of the label codebook. For notational simplicity, elements of matrices $\mathbf{P}_{l,c}$ and $\mathbf{T}_{l,c}$ are denoted in the form $\mathbf{P}(a_i, a_j|l, c)$ and $\mathbf{T}(a_i, a_j|l, c)$, where the first two variables a_i and a_j define matrix element indices of the labels (transition from a_i to a_j), whereas l defines the lag (number of non-specified labels between a_i and a_j, i.e., $l = j\text{-}i$) and c defines the concept (tag number) under consideration.

The first step is to normalize the transition probability from each label to all other labels (right stochastic matrix) by having:

$$P'(a_i,a_j \mid l_d,c_k) = \frac{T(a_i,a_j \mid l_d,c_k)}{\sum_{x=1}^{N_q} T(a_i,a_x \mid l_d,c_k)} \tag{1}$$

where N_q is the label codebook size. Next, the probability that a specific transition occurs during the presence of a tag instead of all other possible transitions is *added cumulatively* to $P'_{l,c}$:

$$P''(a_i,a_j \mid l_d,c_k) = P'(a_i,a_j \mid l_d,c_k) + \frac{T(a_i,a_j \mid l_d,c_k)}{\sum_{x=1}^{N_q}\sum_{y=1}^{N_q} T(a_x,a_y \mid l_d,c_k)} \tag{2}$$

Finally, the probability that a specific transition occurs during the presence of a concept c_k instead of any other concepts is incorporated in the final activation matrix $\mathbf{P}_{l,c}$ by normalizing values over all possible tags, i.e., having:

$$P(a_i,a_j \mid l_d,c_k) = \frac{P''(a_i,a_j \mid l_d,c_k)}{\sum_{z=1}^{N_c} P''(a_i,a_j \mid l_d,c_z)} - \frac{1}{N_c} \tag{3}$$

In other words, the cumulative probability of a transition from a_i to a_j in the case of tag c is divided by the sum of probabilities of the same transition occurring during all possible tags $\boldsymbol{c}$. If a transition becomes equally probable for all concepts, therefore containing no information value, it would have a probability of $1/N_c$. Therefore, each element in all matrices has $1/N_c$ subtracted from its original value in order to have zero activation for the fully random case and a negative value for transitions that occur on average more often during other concepts.

Now each matrix $\mathbf{P}_{l,c}$ keeps a record of normalized transition probabilities from label $s[t\text{-}l]$ to $s[t]$ in the input sequence $\boldsymbol{s}$ when an external information source, called concept c, is activated. Since values of $\mathbf{P}$ are not classical probabilities in the range [0, 1] due to a three-stage normalization process, values of P will be referred to as activation values and $\mathbf{P}$ will be referred as an activation matrix.

2.2 Recognition

During recognition, label transitions in a new input sequence are used as pointers to the activation matrices $\mathbf{P}$. The activation level of a concept c_i at time t given a new input sequence $\boldsymbol{s}$ can be computed by summing over the transition probabilities at different lags, expressed mathematically as:

$$A(c_i,t) = \sum_{d=1}^{N_l} P(s[t-l_d],s[t] \mid l_d,c_i) \tag{4}$$

In order to do pattern recognition, this activation is computed in parallel for all concepts c_i that are included in the search space in order to see what concept is most likely given the present input. This provides a temporally local activation estimate for each concept candidate. However, in many applications it is useful to examine the activation output in a larger temporal window since the events that are being recognized spread over several subsequent time frames. One possible way to do this is to first low-pass or median filter the activation curves in a larger temporal window. In speech recognition experiments the best results were obtained by recursively accumulating the activation level frame by frame with a decay factor λ (5), and then filtering the outcome with a median filter.

$$\begin{aligned} \hat{A}(c_i,t) &= A(c_i,t) + \hat{A}(c_i,t-1) - \frac{\hat{A}(c_i,t-1)}{\lambda} \\ &= A(c_i,t) + \hat{A}(c_i,t-1)(1-\frac{1}{\lambda}) \end{aligned} \tag{5}$$

Once temporal filtering has been performed, a winning concept c_i for each time frame is chosen by selecting the one with the highest activation level. For speech recognition, a median filter of 250 ms and $\lambda = 6$ were found to be effective.

It should be noted that the algorithm can be run in parallel for several input streams in order to incorporate several sources of information (e.g., prosody features or some other contextual data). This transforms frequency and activation matrices into the form $\mathbf{T}_{\psi}(a_i,a_j|l,c)$ and $\mathbf{P}_{\psi}(a_i,a_j|l,c)$, where Ψ denotes the number of the input stream being processed. Training is performed similarly to the single stream condition in order to build separate concept matrices for each concept at each lag and for each stream. In the testing phase, the probability output from all streams is combined to have a probability of a concept c_i at time t of:

$$A(c_i,t) = \sum_{\psi=1}^{\|\psi\|} \left(\sum_{d=1}^{N_l} P_{\psi}(s[t-l_d],s[t] \mid l_d,c_i) \right) * \omega_{\psi} \tag{6}$$

where ω_{ψ} is a weighting factor defined for each input stream. Since only the transitions that are informative in relation to a specific concept receive values above zero, the inclusion of additional streams should not bias or degrade the recognition process. However, no such parallel stream processing was used in the experiments reported in this paper.

3. Experiments

3.1 Material and evaluation

The material used in the experiments consisted of the TIDIGITS corpus [9] that contains continuously spoken digit sequences (1-7 digits per utterance) in different dialects of American English by 225 different speakers (111 males, 114 women, f_s = 16 kHz). Training material consisted of the original male and female training set of TIDIGITS, except for the noise experiments that were trained using only male data (N = 4235 utterances) due to practical issues. Test data consisted of the full test set, except for noise experiments where N = 650 utterances were chosen randomly from speakers in the test/male set.

Audio signals were converted into label sequences using vector quantization. MFCCs (N = 12 coefficients) were extracted every 10 ms using a 20 ms Hamming window. In addition, speech was segmented into phone-like segments using a blind segmentation algorithm [10] and segmental MFCCs were extracted using features only from segment center points. A *k-means* VQ codebook of size N_q = 150 was created from a subset of the training data using only segmental MFCCs and the Euclidean distance as a distance metric. All utterances were then quantized using the obtained codebook with the full temporal resolution of MFCC vector every 10 ms. Concept tags related to each utterance were extracted directly from the signal annotation, one for each digit, yielding a total of N_c = 11 different tags including "*oh*" and "*zero*". As an outcome, each utterance was described as one VQ-sequence and an unordered set of tags related to the words in the utterance.

Evaluation was performed by having the algorithm provide an ordered set of N words, where N was the true number of words in each utterance. The word hypotheses were the N most activated models (cumulative sum over an entire utterance), and their temporal location was defined as the point where the mean of their cumulative activation sum was reached. This is a simplification of a real speech recognition task, but a necessary one since the current implementation does not have an activity based decoding mechanism for word strings, i.e., it does not know whether a very brief but large activation of a model can be a target event being recognized. This type of knowledge about word string length has been shown to increase the HMM recognition rate by approximately 2 % [11] in the same task in noise. Since a lack of temporal decoding leads to an inability to differentiate between subsequent repetitions of a single word and a long pronunciation of the same word, utterances with repetitions of a same word (like "*six-six-nine-two*") were excluded from the test set.

3.2 Results

Development of the algorithm has shown that for speech, results can be improved by extending the lags from zero up to 250 ms, and this was confirmed with the test material. Increases beyond 250 ms do not seem to have a significant effect. Therefore, lags from 10 ms up to 250 ms were used in the experiments. A median filter of 250 ms and $\lambda = 6$ were used in the post-processing of concept activations. With these settings, clean speech word recognition accuracy in the digit recognition task was 94.34 % for the TIDIGITS test set.

An example of the recognition process is shown in fig. 1, where the utterance "*three-four-one-two-six*" is being analyzed.

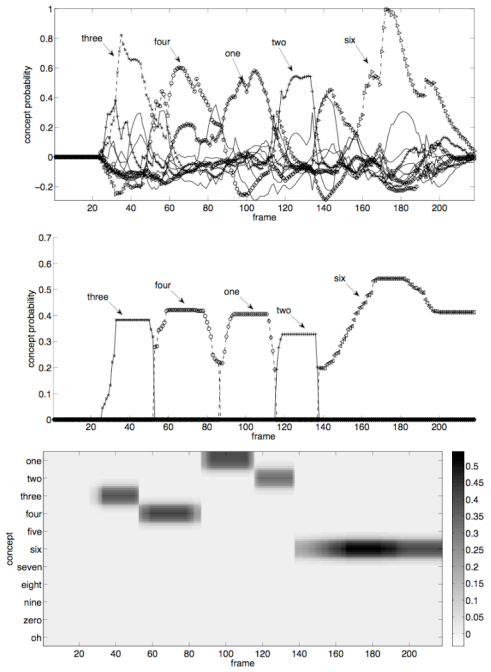

Figure 1: *Top: cumulative activation curves of all 11 recognizers in recognition of utterance "three-four-one-two-six". Middle: activation after median filtering and inhibition. Bottom: Association Response Table (ART) showing the activations with concepts at different rows.*

Activation curves after cumulative activation (5) (top) and median filtering (middle) are shown. Also, a very convenient way to visualize concept specific activations is shown at the bottom. We call this an *Association Response Table* (ART).

As can be seen from the top figure, very salient activations of correct models emerge during the presence of the target word. Other models sharing sub-word structure with the correct words also gain activations temporarily, whereas activations of non-related parts are below zero. Additionally, the temporal boundaries between concepts (fig. 1, middle) seem to provide an accurate word and/or morpheme segmentation of the input. This was noted by a manual inspection using English and Finnish continuous speech.

Noise robustness was tested using white Gaussian noise (WGN) and non-stationary factory noise taken from NOISEX [12]. Figure 2 displays the results as a function of SNR. The effect of WGN is shown separately for a clean training condition, where VQ-codebook and CM models are trained with clean signals, and for a matched condition, where training is performed in similar noise conditions as testing. The CM algorithm performs relatively well at a SNR of 20 dB_{seg} in all cases, but recognition starts to degrade with an increasing rate below that level. However, the recognition rates at low SNRs are still very well comparable to the results reported with continuous density HMMs, without and with noise compensation (e.g., [13,14]), although the task here is simplified since the correct number of words is known beforehand. As expected, once the noise conditions in training and testing are matched, the recognition rate is significantly better and is still nearly 75 % at a SNR_{seg} of 0 dB in WGN. Figure 3 shows an example of speech corrupted by factory noise (SNR = 0 dB) where recognition has still been successful.

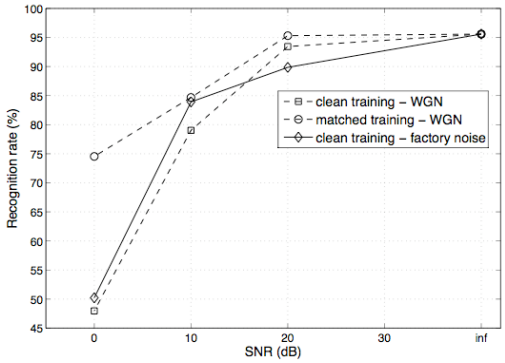

Figure 2: *Recognition in noise. Training performed with clean speech (dashed line with squares) and noisy speech (dashed line with circles) are shown separately for white noise. The effect of factory noise is shown with a solid line.*

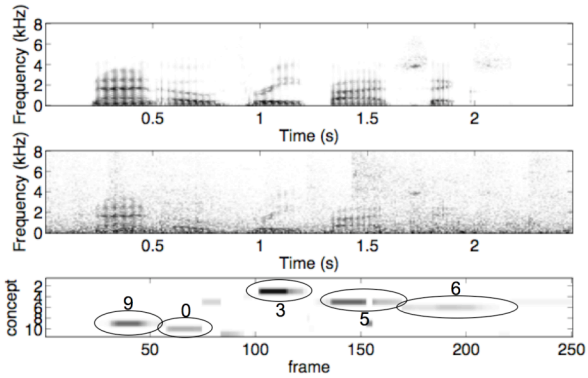

Figure 3: *Top: Logarithmic spectrogram of the clean utterance "nine-oh-three-five-six". Middle and bottom: corresponding spectrogram and CM output with a factory noise of 0 dB* SNR_{seg}.

5. Conclusions

It was shown that given a set of discrete unit sequences and a set of informative tags for each sequence, the CM algorithm is able to create structural models that associate the presence of a specific tag with specific parts of the time-series. This model can be used for recognition of similar patterns in future input, and can handle large amounts of variability and noise in the sequences. This is since CM does not make the Markov property assumption, i.e., it does not attempt to pack all the information of the past states to the current state of the system, but instead integrates information over larger temporal windows. This makes it robust against local distortions in the input.

From the perspective of speech recognition, CM is able to learn statistical models for separate words from continuous spoken language and recognize them with high accuracy. Noise robustness of CM is also comparable to existing noise-compensation approaches used in HMM algorithms [13,14], although it does not employ any kind of special mechanism for dealing with noisy input. However, since CM in its basic form lacks a mechanism for detecting and decoding speech specific word-like units, the recognition task was slightly simplified. Development of such a decoding mechanism tailored especially for word or phone recognition would bring the system closer towards real speech recognition applications where the number of words cannot be assumed beforehand.

From a computational complexity perspective the algorithm is efficient. This is especially true for recognition, where post-quantization steps only include the retrieval and summation of activation values from memory and the temporal filtering of the obtained activation curves. However, the memory requirements for storing activity and frequency matrices may become problematic with a large number of concepts or with very large codebooks. Still, the statistics are very sparse and the memory requirements can be reduced with proper compression.

Finally, it is important to note that the learning and recognition processes are purely incremental, making it possible to perform recognition and further learning simultaneously all in real time. This also opens up new possibilities for further improvements of the algorithm, e.g., for reinforcement learning and refinement of the models based on feedback from recognition.

Acknowledgements

This research is funded as part of the EU FP6 FET project Acquisition of Communication and Recognition Skills (ACORNS), contract no. FP6-034362.

References

[1] Gales, M., and Young, S., "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends in Signal Processing, Vol. 1, No. 3, pp. 195-304, 2008

[2] Parveen, S., and Green, P. D., "Speech Recognition with Missing Data using Recurrent Neural Nets", Proc. Advances in Neural Information Processing Systems, NIPS*2001, 2001

[3] Van hamme, H., "HAC-models: a Novel Approach to Continuous Speech Recognition", Proc. Interspeech, Brisbane, Australia, 2008

[4] Van hamme, H., "Integration of Asynchronous Knowledge Sources in a Novel Speech Recognition Framework", ISCA Tutorial and Research Workshop (ITRW), Aalborg, 2008

[5] Ten Bosch, L., Van hamme, H., and Boves, L., "A Computational Model of Language Acquistition: Focus on Word Discovery", Proc. Interspeech, Brisbane, Australia, 2008

[6] Gold, K., Doniec, M., Crick, C., and Scasselati, B., "Robotic vocabulary building using extension inference and implicit contrast. Artificial Intelligence, Vol. 173, pp. 145-166, 2009

[7] Roy, D., "Semiotic Schemas: A Framework for Grounding Language in Action and Perception", Artificial Intelligence, Vol. 167, No. 1-2, pp.170-205, 2005

[8] Räsänen, O., Laine, U. K., and Altosaar T., "Computational language acquisition by statistical bottom-up processing", Proc. Interspeech'08, pp. 1980-1983, 2008

[9] Leonard, R. G., "A Database for Speaker-Independent Digit Recognition", Proc. ICASSP 84, Vol. 3, p. 42.11, 1984

[10] Räsänen, O., "Speech segmentation and clustering methods for a new speech recognition architecture", M.Sc. thesis, TKK, Finland, 2007

[11] Kim, J., Haimi-Cohen, R., and Soong, F., "Hidden Markov Models with Divergence Based Vector Quantized Variances", Proc. ICASSP'99, pp. 125-128, 1999

[12] http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html

[13] Renevey, P., "Speech Recognition in Noisy Conditions Using Missing Feature Approach", Ph.D. thesis, Lausanne, EPFL, 2000

[14] Raj, B, and Stern, R. M., "Missing-Feature Approaches in Speech Recognition. IEEE Signal Processing Magazine, Vol. 22, pp. 101-116, 2005