Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A method for noise-robust context-aware pattern discovery and recognition from categorical sequences

Okko Räsänen*, Unto K. Laine

Aalto University, School of Electrical Engineering, Department of Signal Processing and Acoustics, P.O. Box 13000, FI-00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 23 June 2010
Received in revised form
10 February 2011
Accepted 11 May 2011
Available online 18 May 2011

Keywords:
Speech recognition
Pattern discovery
Categorical sequence analysis
Weakly supervised learning
Pattern recognition

ABSTRACT

An efficient method for weakly supervised pattern discovery and recognition from discrete categorical sequences is introduced. The method utilizes two parallel sources of data: categorical sequences carrying some temporal or spatial information and a set of labeled, but not exactly aligned, contextual events related to the sequences. From these inputs the method builds associative models able to describe systematically co-occurring structures in the input streams. The learned models, based on transitional probabilities of events observed at several different time lags, inherently segment and classify novel sequences into contextual categories. Learning and recognition processes are purely incremental and computationally cheap, making the approach suitable for on-line learning tasks. The capabilities of the algorithm are demonstrated in a keyword learning task from continuous infant-directed speech and a continuous speech recognition task operating at varying noise levels.

1. Introduction

What makes a system or agent behave intelligently? We all probably agree that, at least, it should be able to learn from experience and it should even learn associations and structures between multiple data streams it is exposed to. Learning means that the system has sensors to collect multimodal data, efficient mechanisms to process and refine the acquired information, and a memory where to accumulate experiences in the form of compact, dynamic, models. The created models of reality should be easily accessible and become activated in comparable situations to those where they once emerged. The memory should support continuous, active pattern discovery, and recognition, that form a necessary basis for learning of new skills and for improving those that have been adopted earlier. In terms of artificial intelligence, there is need for new methodologies that enable efficient learning and extraction of important structures from noisy and sparse data and, especially, that can learn under minimal supervision.

One exemplar of efficient learning is the human being. We utilize a number of senses in our behavior and we are able to deal with continuous streams of several parallel inputs they provide. We discover and memorize *associations* across different sensory modalities, our actions, and the internal states of our system in a process called grounding. In other words, we are able to form conceptual representations of the surrounding world by being sensitive to patterns that occur in specific contexts. Once a familiar pattern is perceived, the associative links, and thereby the meaning of the event, become activated and the input is recognized. This type of multimodal or contextual *pattern discovery* process differs from traditional *pattern recognition* in a way that external expertise is not needed or it is minimal during the learning of patterns. Classically, the expertise manifests itself as careful preparation of the training data, tweaking of the initial model parameters to fit better to the present situation, or as using a task-specific architecture to solve the problem. However, such a priori knowledge is not always available, or its usage may not be ecologically plausible, e.g., when studying self-learning systems.

Many learning tasks can be perceived as weakly supervised pattern discovery problems in the absence of explicit teaching. Weak supervision refers here to learning conditions where the training samples are not explicitly segmented and aligned to pinpoint their belongingness to a specific target category. Instead, a number of possible contextual variables (target classes) are presented in parallel with the input, but it is not known whether all these classes are present in the data and, if they are, where they are (see, e.g., [1–3]). The task of the learning algorithm is then to discover those patterns from the data that are relevant for each contextual variable.[1] For example, human infants have to

* Corresponding author. Tel.: +358 9 47022499; fax: +358 9 460 224.
E-mail address: okko.rasanen@aalto.fi (O. Räsänen).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.05.005

[1] Mutual dependencies of two or more data types have been widely studied in the field of information retrieval. However, methods such as Latent Semantic Indexing (LSI; [6]) and Supervised Semantic Indexing [7] concentrate on the modeling of distances between individual, static, and relatively high-level objects between two or more data types (e.g., search term and document relationships). For example, the heart of the LSI is its ability to estimate semantically reasonable co-occurrence probabilities for object pairs that have never actually co-occurred in

solve the word-referent problem between acoustically perceived word forms and a finite set of possible objects or events that the words are related to without teaching by explicit object naming (e.g., [4]). The same is true for applications of process quality monitoring in process industry or in automated medical diagnostics, where large amounts of sensory data might be available and the task is to find patterns associated with a set of normal and defective conditions. The benefits of effective recognition of events in these cases do not need further elaboration. It has also 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., [5]). However, the existing methodology for weakly supervised, or context-aware, pattern discovery is far from complete. Even the theories of pattern discovery and associative learning are struggling with the definitions of patterns themselves, and often avoiding this issue by either abstracting the notion of a pattern to a symbolic level or by defining a pattern as a set of signal properties relevant for the given classification task (this topic is discussed further in [8] and references therein).

This paper introduces a novel approach for associative pattern discovery from sequential data. This method, called the concept matrix (CM) algorithm was originally developed as a learning mechanism for a weakly supervised language learning task, where a computational agent ("infant") interacts with a virtual caregiver and attempts to learn multimodal associations between acoustic and visual input, i.e. attempts to learn language (see [9] for an overview of the ACORNS project). This is a difficult task, since the learner cannot be assumed to have innate linguistic knowledge and therefore it does not even know how to segment words from speech in the beginning. In practice, this means that there are no pointers to the parts of the signal that are meaningful to the task, and it is not known how these parts are related to other sources of information like the visual context (word-referent problem; see, e.g., [4]). In order to solve the task, the CM combines information from two input streams and finds co-occurrence relations between them. It learns recurring structures in similar contexts and recognizes them from new input. Contrary to the hidden Markov models (HMMs) that are the state-of-the-art in speech recognition and widely used in many other pattern recognition tasks, the CM does not make the Markov property assumption regarding independence of the subsequent states. This makes it capable of finding structures between non-adjacent events and robust against temporally local distortions. The algorithm takes a functional stance to the definition of a pattern, defining pattern as a collection of spectrotemporal events that are associated to a specific context or set of contexts. The patterns can manifest themselves in many different ways, but are considered as equal as long as their predictions regarding the contextual state converge.

The paper is organized as follows. Section 2 reviews the properties of temporal patterns and related limitations of the existing pattern recognition methodology, especially Markov models and their extensions to higher orders. Section 3 presents the CM algorithm in detail. In Section 4, the performance of the CM algorithm is demonstrated in two slightly different weakly supervised word learning tasks using continuous. Finally, conclusions are drawn in Section 5.

While the experiments are concentrated around pattern discovery from speech due to the field of expertise of the authors, the algorithm is not theoretically limited to audio processing and can be utilized for any kind of context-aware pattern discovery from time-series that can be expressed as discrete sequences either by quantization or clustering of signal segments (see [10]).

1.1. Notation and definitions

This study is focused on discovery and modeling of patterns and structures in discrete sequences. A sequence X is generated by a source of finite alphabet $\mathbf{A}$ with cardinality $|\mathbf{A}|=N_\mathrm{A}$. An element of X at time t is expressed by X_t, and its particular value by $X_t=a_t$, where $a_t\in\mathbf{A}$. The probability of an event $X_t=a_t$ is given by $P[X_t=a_t]$ and the conditional probability by $P[X_t=a_t|X_{t-1}=a_{t-1}]$ when the event at t depends on the previous event $X_{t-1}=a_{t-1}$. In CM, each sequence is assumed to occur in context $\boldsymbol{c}\in\mathrm{C}$, $\boldsymbol{c}=\{c_1, c_2, \ldots, c_n\}$, i.e. an unordered set of context variables. Successful learning produces links that associate specific structures in X to specific contexts $\boldsymbol{c}$. These associations between patterns and contexts are called *concepts.*

2. Temporal patterns and the Markov property

For many types of real world data, the patterns of interested extend over time or space. However, the methods applied to pattern analysis cannot extract these patterns as a whole without a priori knowledge regarding their structure. This forces the analysis to be performed at a proper level of granularity where all necessary details are preserved in order to differentiate between pattern classes, but where it is still possible to perform classification in a computationally feasible way and with the available finite and possibly sparse training data. For digital signal processing this means that the signal has to be divided into small discrete segments, or frames, and transformations are applied to represent the signal in each frame using a set of descriptive features. It is clear that the finer the temporal slicing, the more temporally inter-dependent frames are produced. The estimation of the strength of inter-dependencies between frames of different kinds and at different distances is the main issue of all pattern discovery algorithms. Element inter-dependencies should be stronger within a pattern than between different patterns. In order to perform pattern discovery on such time-series of data, efficient methodology is required for analyzing and modeling of these structures.

If the time-series data can be converted into a sequence of discrete elements, one method to estimate the amount of structure at different temporal distances in categorical sequences is to use the information theoretically motivated mutual information function (MIF; [11–12]):

$$M(k)=\sum_{i,j}P(a_i,a_j|k)\log_2\left(\frac{P(a_i,a_j|k)}{P(a_i)P(a_j)}\right) \tag{1}$$

where $P(a_i,a_j|k)$ denotes the probability of an element pair a_i and a_j in the sequence, separated by lag k. In principle, the MIF measures the average amount of information (in bits) known about the future state of the sequence at distance k.

Examples of patterns that span across several analysis frames (or pixels) can be found in audio and images. Speech, for example, has inherent dependencies that can span even across entire utterances. Fig. 1 shows the MIF computed from a large set of high-quality, continuous, English speech waveforms that have been windowed, transformed to the spectral domain features (frame by frame), and vector quantized (VQ) with a codebook of size $N_\mathrm{A}=150$. The windowing was produced using 10 ms frame shifts that are commonly used in automatic speech recognition (ASR), yielding approximately one million frames in total. The figure illustrates that the dependencies between the elements in the VQ sequences are not limited to subsequent frames, but

(*footnote continued*)
the training data. In contrast, the method proposed in this work focuses on the discovery of *a priori unknown* and *temporally* or *spatially distributed patterns* from sparse data that have systematical relationships to specific contexts.

degrade gradually as the temporal distance between the frames increases (the curve is actually very close to linear on the log–log scale). From the perspective of modeling and prediction, the MIF curve suggests that it may be useful to incorporate information across several preceding frames in order to predict the future unless the source is truly Markovian.

One challenge in the modeling of temporally distributed long-range dependencies is the fact that, despite these dependencies, the variability in the signal may be so high that patterns never recur in exactly the same form (i.e. data sparsity problem). Another issue is that there is often no way to ensure that *all* subsequent frames are representative of the patterns that we are interested in. This may be due to the feature extraction process or characteristics of the source that causes some of the frames to carry more reliable information than others, or there may be significant amounts of noise in the signal. It is also possible that the meaningful patterns are interleaved in the data stream. This means that modeling of patterns as rigid n-tuple prototypes or even as collections of n-tuple exemplars is rarely feasible, but more sophisticated methodology is required.

2.1. Markov models

Traditionally, Markov chains and HMMs have been applied to a great variety of pattern recognition tasks on temporally or spatially evolving data, e.g. speech recognition [13], image recognition [14–15], and protein sequence analysis [16] in which they often have been considered relatively successful. For example, an HMM-based speech recognizer could easily be applied to recognition of words from the above speech VQ data with good results, especially if labeled training data were available [17,18].

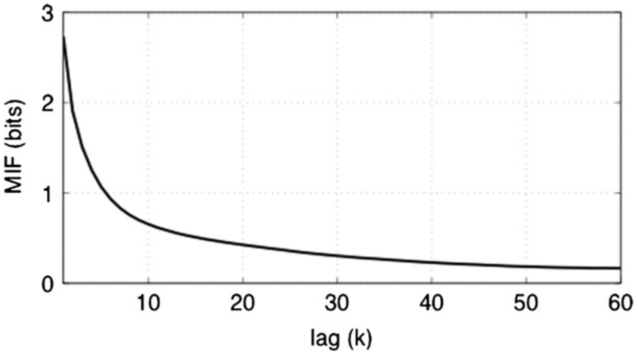

Fig. 1. Mutual information function for vector quantized speech (codebook size 150). One lag unit corresponds to a temporal distance of 10 ms.

However, Markov models share one very essential limitation, namely the *Markov property* itself. More precisely, the Markov process assumes that *the likelihood of a future state depends only on the present state and not on any past states.* The first-order Markov chain is given by

$$P[X_t = a_t | X_{t-1} = a_{t-1},\ X_{t-2} = a_{t-2}, \ldots, X_{t-k} = a_{t-k}] = P[X_t = a_t | X_{t-1} = a_{t-1}] \tag{2}$$

This Markov assumption does not hold for many types of data, including most features extracted from auditory or visual input (see Fig. 1). Fig. 2 illustrates the problems associated with the first-order Markov chains. The left panel in Fig. 2 shows the root-mean-square error (RMSE) of all transition probabilities at distance k between stochastic matrices $\mathbf{P}_1^k$ derived from first-order Markov chain $\mathbf{P}_1$ (the probability of moving from element a_i to a_j in k steps) and the corresponding $\mathbf{P}_k$ matrices estimated directly from the data by collecting lagged bi-gram statistics by always skipping $k-1$ elements between a_i and a_j. If the Markov assumption were to hold, the error should be minimal. Since speech is neither a Markov process nor stationary at the level of acoustics, the Markov chain yields incorrect transitional probabilities for lags greater than $k=1$ and the difference between the real estimate and the Markov process grows quickly at small lags. When k becomes sufficiently large, the difference starts to decrease because the amount of structure at larger distances diminishes, i.e. the transition probabilities estimated directly from the data approach the overall distribution of elements in the VQ data. The right panel in Fig. 2 illustrates the standard deviation of transition probability values in the case of the first-order Markov process ($\mathbf{P}_1^k$) and for transition probabilities estimated directly from the data. The differences between these two cases are very large in the lag range between 3 and 20, indicating significant structural differences between the models.

The problems of the Markov assumption are acknowledged in the ASR community [13,19], where all the state-of-the-art speech recognizers are based on HMM architectures. The existing solution is to include derivatives (and derivatives of derivatives, i.e. delta and delta–delta coefficients) to local mel-cepstrum features. This is a very important extension that takes into account dynamical changes in the speech spectrum over several subsequent frames. In other words, the speech recognition systems based on Markov models are essentially built on features that fail to conform to the Markov premises! Use of these derivative features has a significant impact on HMM-based ASR performance (see, e.g., [20]).

In order to overcome the limitations of studying transitions only between two subsequent states, higher order Markov chains can be used. Higher order Markov chains include k preceding

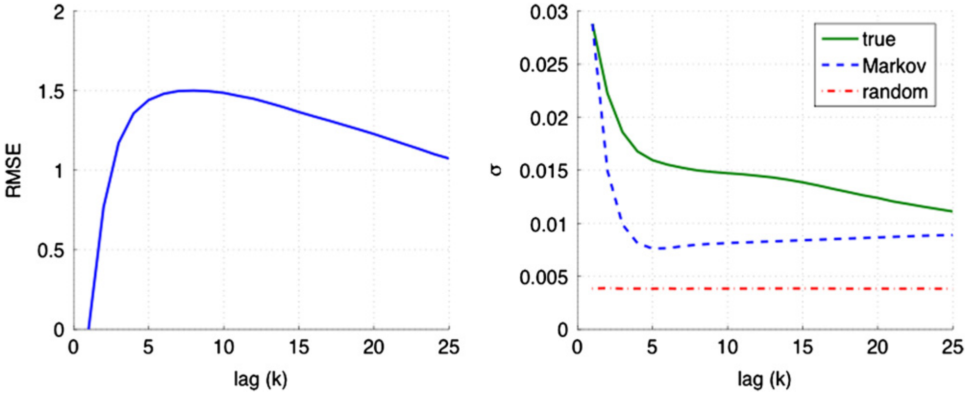

Fig. 2. *Left*: root-mean-squared error (RMSE) between a Markov chain model and transition probabilities estimated directly from the speech VQ data. *Right*: standard deviation of transition probability values for different lags of the Markov chain model, true estimates and the purely randomized case.

(consecutive) states in the computation of the next transition:

$$P[X_t = a_t | X_{t-1} = a_{t-1}, \ldots] = P[X_t = a_t | X_{t-1} = a_{t-1}, \ldots, X_{t-k} = a_{t-k}] \quad (3)$$

However, the problem with the kth order Markov chain is that it requires $(m-1)m^k$ parameters with m possible states. For most applications, sparsity of the training data limits the maximal order of the model to k=2–4. In addition, decoding of high-order HMMs is computationally expensive (see, e.g., [21] for discussion and possible solutions).

Naturally, possible solutions for extending the HMMs to higher orders have been studied. Casar and Fonollosa [22] extended first-order HMMs to incorporate third-order Markov chains in the acoustic models of a speech recognizer by pruning away low-frequency n-grams. Kobayashi et al. [23] modified the second-order Markov model by pairing each hidden state with a visible state. Lee and Lee [24] approximated high-order Markov chains by replacing the history of states with an additional parameter that models the duration spent in the previous state. All these approaches yielded improvements in HMM speech recognition accuracy by relaxing the assumptions of independency and/or piecewise stationarity of the states.

Possibilities to overcome the parameter estimation problem of high-order Markov models have been also studied outside the field of speech recognition. Raftery [25] has devised a model called *mixture transition distribution* (MTD) where Eq. (3) can be compressed into a linear combination of one matrix $\mathbf{Q}$ with weights $\boldsymbol{\lambda}=\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ for different lags:

$$P[X_t = a_t | X_{t-1} = a_{t-1}, \ldots, X_{t-K} = a_{t-K}] = \sum_{k=1}^{K} \lambda_k Q(a_{t-k}, a_t) \quad (4)$$

where $\sum \lambda_k = 1$ and (a_{t-k}, a_t) denotes transition the from a_{t-k} to a_t. In [26] examples of cases are given where the MTD improves the estimation accuracy from Markov chains of the same order, despite the fact that the MTD approach reduces the number of parameters to $k+m^2$. However, the MTD requires solving a highly nonlinear optimization problem in order to obtain maximum likelihood-estimate values for $\mathbf{Q}$ and $\boldsymbol{\lambda}$ [26,27]. Therefore the MTD has been modified to allow $\mathbf{Q}$ to change as a function of lag, which makes parameter estimation much easier [27,28]:

$$P[X_t = a_t | X_{t-1} = a_{t-1}, \ldots, X_{t-K} = a_{t-K}] = \sum_{k=1}^{K} \lambda_k \mathbf{Q}_k(a_{t-k}, a_t) \quad (5)$$

This generalization is referred to as MTDg [28]. Although this increases the number of parameters from the standard MTD, the model is still parsimonious when compared to the Markov chains of the same order. MTDg can also represent structures that MTD cannot due to its increased independence of transitions at different lags [29]. However, the simultaneous estimation of K values of λ_k and K matrices $\mathbf{Q}_k$ is still time consuming with the EM-algorithm and the global maximum of log-likelihood is not guaranteed.[2] Therefore Ching et al. [27] have suggested a straightforward method for estimating transition probabilities $\mathbf{Q}_k(a_i,a_j)$ directly from the frequency of transitions f_k at lag k in the sequence:

$$Q_k(a_i,a_j) = \frac{f_k(a_i,a_j)}{\sum_{j=1}^{m} f_k(a_i,a_j)} \quad (6)$$

i.e., in their method, a stochastic matrix is created in parallel for each lag by computing relative frequencies of all transitions in the training data. This is computationally much more effective than iterative EM-algorithm based estimation (cf., [30]) and is therefore suitable also for tasks that require incremental learning or low computational complexity during training. Ching et al. [27] demonstrated the efficiency of the approach in the prediction of sales demand data and web browsing behavior. The earlier work of Saul and Pereira [31] also suggests the use of Eq. (6) for estimation of initial parameters of Mixed-order Markov model that are then refined iteratively using the EM-algorithm.

As will be seen in the next section, the proposed concept matrix algorithm builds on the idea of parallel transition matrices for each lag but relaxes the requirements of the estimation of λ, which can be consuming for large number of lags [27], but also unnecessary for weakly supervised pattern discovery and classification. By introducing an additional discriminating normalization procedure to the collected statistics, the obtained high-order models can be used for efficient modeling of long-range dependencies that can be directly applied to pattern recognition with one-pass training. To our knowledge, this is the first approach to the modeling of long-range dependencies as mixtures of bi-gram statistics to the problem of weakly supervised pattern discovery and classification, whereas previous approaches [24–32] concentrate solely on the fundamentally different task of sequence prediction.

3. The concept matrix algorithm

The concept matrix (CM) method was developed as a part of studies related to computational modeling of a self-organizing agent that could reflect basic pattern discovery and recognition skills from speech input (an EU FP6 project called ACORNS; [9]). The fundamental idea behind the CM algorithm is the desire to *learn associations* between temporal, spatial, or spatio-temporal patterns and some contextual information. It is based on the assumption that recurring temporal or spatial structures exist in the domain of the first categorical data stream, or *primary sequence* (PS; e.g., vector quantized speech or pixels), and that these structures are somehow related to the second data stream, or *contextual input* (CI; e.g., visual objects or system states), which is simultaneously generated by a categorical information source. The data rate of the PS is typically higher than that of the CI and, secondly, CI is typically qualitatively on a higher level than PS. If PS represents a *temporal sequence* (TS), the corresponding CI may give contextual data in the same temporal order (in the form of another sequence) or it just gives an unordered set of concepts related to a certain TS.

The strength of the proposed CM algorithm is that it does not require an exact alignment of the events occurring in different input streams, and it can deal with large amounts of variability and noise at the input. It learns to extract meaningful segments of the signal simply by accumulating evidence across different contexts. Instead of making the Markov assumption, the algorithm models transitions, i.e. statistics of element pairs, at different distances (lags) *in parallel* similarly to the MTDg. This leads to a faithful representation of signal structure up to the desired lag (temporal span or spatial distance) and avoids the estimation of a huge number of parameters employed by high-order Markov models. The approach also circumvents the data sparsity problem, since the ratio of free parameters and number of exemplars in the training data stays constant for any number of lags (there are always $L-k+1$ transitions at lag k in data of length L). For a precise mathematical treatment of the limits of cross-situational learning, i.e., how learning rate and recognition accuracy are constrained by the number of (ideally identifiable) patterns and contextual variables, an interested reader is suggested to see [33]. Here we simply adopt the assumption that a pattern can be learned if

[2] In its most general form, MTDg actually has $k+\sum_{m=2}^{k-1}(k/m)$ different $\mathbf{Q}$ matrices if joint probabilities of several lagged periods are combined, e.g. $P(X_t=a_0|X_{t-2}=a_2, X_{t-1}=a_1)$. However, these models are generally overparametrized for practical use [26].

occurs in a specific context above chance level, where chance can be derived from the mean of occurrence frequencies of the pattern over all contexts.

In the following, the CM method is derived step by step and some interesting properties of the formulation are discussed.

3.1. Inputs

The TS data consists of sequences with elements a_i from a finite alphabet $\mathbf{A}$. In this paper, the TS elements consist of VQ indices derived from audio waveforms by vector quantization of spectral representations. The CI information source is represented by a set of so-called context tags $C=\{c_1, c_2, \ldots, c_n\}$. These tags are integer values that represent invariant outputs of another process that are concurrently activated with the input sequence (e.g., a categorization process performed in another modality or some other group of manually defined events that should be associated with the structures in the input sequence; see also [34]). The CI information does not always have to be correct, and it may contain insertions and substitutions that do not fully correspond to the contents of the TS stream.

3.2. Training

When a subset of context tags $\boldsymbol{c} \subseteq C$ and a related sequence $X=[a_t, a_{t+1}, \ldots, a_{t+m}] \in \mathbf{A}^m$ are represented, the algorithm starts to collect frequency data regarding the occurrences of element pairs in the sequence X at lags $\boldsymbol{k}=\{k_1, k_2, \ldots, k_K\}$. Transitions at lag k-means that the subsequent $k-1$ elements are always skipped when collecting transition statistics (i.e., a transition from a_t to a_{t+k}). This frequency data is stored in matrices $\mathbf{F}_{k,c}$ specified by the lag k and the context $\mathbf{c}$. A separate matrix exists for each context tag at each lag, yielding a total of KN_C matrices of size $N_\mathrm{A} \times N_\mathrm{A}$, where N_C is the total number of all possible context tags, K is the number of used lags, and N_A is the size of the alphabet. The original elements can be used as pointers to $\mathbf{F}$ when the number of occurrences of the corresponding element pair is required.

After the frequency data from all $X \in \mathbf{X}$ in the training set are collected, data from every $\mathbf{F}_{k,c}$ are normalized to activation matrices $\mathbf{Q}_{k,c}$. For notational simplicity, elements of any matrix $\mathbf{F}_{k,c}$ are denoted in the form $\mathbf{F}_k(a_i,\ a_j|c)$, where the first two variables a_i and a_j refer to the matrix element of the corresponding transition, k defines the lag (number of non-specified elements between a_i and a_j, i.e. $k=j-i$), and c refers to the context in which the transition takes place (tag index).

The first step in normalization is to compute the transition probabilities from each element to all other elements (right stochastic matrix $\mathbf{P}^\mathrm{S}$) at each lag by having:

$$\mathbf{P}_k^\mathrm{S}(a_j|a_i,c) = \frac{\mathbf{F}_k(a_i,a_j|c)}{\sum_{j=1}^{N_A} \mathbf{F}_k(a_i,a_j|c)} \tag{7}$$

Since the idea is to classify novel inputs into one of the existing categories, the probability that a specific transition occurs during the presence of a context c_n instead of any other contexts is incorporated in activation matrix $\mathbf{P}^\mathrm{C}$. The conditional probability[3] $P(a_i,a_j \in c_n|k)$ is given by

$$\mathbf{P}_k^\mathrm{C}(a_j|a_i,c_n) = \frac{\mathbf{P}_k^\mathrm{S}(a_j|a_i,c_n)}{\sum_{m=1}^{N_\mathrm{C}} \mathbf{P}_k^\mathrm{S}(a_j|a_i,c_m)} \tag{8}$$

If the sequence X is generated by an independent and identically distributed random process, all the activation values are equal to $1/N_\mathrm{C}$. It is often practical to subtract this constant from the conditional probabilities of Eq. (8) in order to normalize these meaningless values to zero and to have the sum of activations over all possible models c is zero at all times. This final step leads to the activation matrix $\mathbf{Q}_{k,c}$:

$$\mathbf{Q}_{k,c} = \mathbf{P}_{k,c}^\mathrm{C} - \frac{1}{N_\mathrm{C}} \tag{9}$$

To summarize, each matrix $\mathbf{Q}_{k,c}$ keeps a record of normalized transition probabilities from element X_{t-k} to X_t in the input sequence X when an external information source, called context c, is activated simultaneously.

Since the ultimate aim is to assign structurally significant parts of input sequences into one or more previously perceived context categories c, and not to predict the next element in the sequence as in the previously reported approaches using transition mixture models [25–32], the computationally expensive estimation of the lag-specific weights $\boldsymbol{\lambda}$ used in the MTD algorithms (Eq. (5); [26,28]) is not necessary. This is because the Eqs. (7) and (8) already provide the maximum likelihood estimate for context c given transition from a_i to a_j. This can be understood by considering the situation where no a priori knowledge of the pattern lengths is known in advance. If all competing models C share similar assumptions regarding temporal structure of the patterns, any lag-specific weights λ_k would be the same in all models and would diminish in Eq. (8). In the absence of lag-specific weights, MIF computed from the training data (Eq. (1)) is a good way to estimate the number of lags that can potentially affect the pattern discovery performance.

In contrast to the task of *pattern discovery* discussed in this work, the lag-specific weights $\boldsymbol{\lambda}$ become essential in *sequence prediction* where the aim is not to assign perceived patterns into a number of different classes, but to learn probability distributions for future values of the data. Therefore it is suggested to use the original MTD [25,26,28] or its derivatives (such as Mixed-order Markov models in [31,32]) for prediction. This is because the EM-algorithm used to estimate $\boldsymbol{\lambda}$ and $\mathbf{Q}$ parameters in these models aims at maximal sequence prediction likelihood at the cost of rich structural description of long-range dependencies. Such optimization of parameters for prediction only degrades classification performance in weakly supervised pattern discovery tasks.

3.3. Recognition

During recognition, ordered element pairs in a new input sequence are used as pointers to the activation matrices $\mathbf{Q}$. The activation level of a context c_n at time t given a new input sequence X can be computed by summing over the previously learned transition probabilities at different lags, expressed mathematically as

$$A(c_n,t) = \frac{1}{K}\sum_{m=1}^{K} \mathbf{Q}_{k_m}(X_t|X_{t-k_m},c_n) \tag{10}$$

where K is the total number of lags and corresponding stochastic matrices used in the CM-analysis. The normalization term $1/K$ ensures that the activation has a maximum value of $1-1/N_\mathrm{C} \approx 1$ (if $N_\mathrm{C} \gg 0$), and it can be omitted if only the relative activations of concepts are of interest.

The activation $A(c_n,t)$ is computed in parallel for all models c_n that are included in the search space in order to see what model is most likely given the present input. This provides a temporally local activation estimate for each model candidate. However, in many applications it is useful to examine the activation output in larger temporal windows since the events that are being

[3] The relation "is element of c", is not defined formally. Therefore this expression should be read as: elementary structure (or transition) a_ia_j can be *associated* with concept c with a probability P.

recognized spread over several subsequent elements in the sequence. Since independence of subsequent frames cannot be assumed (i.e., $\prod A(t)$ is not feasible), being the reason why several lags are modeled in the first place, a straightforward approach is to low-pass or median filter the activation curves in a larger temporal window. In the word learning experiments reported in this paper, the best results were obtained by recursively accumulating the activation level frame by frame with a decay factor γ, and then filtering the outcome with a median filter:

$$A'(c_n,t) = A(c_n,t) + A'(c_n,t-1) - \frac{A'(c_n,t-1)}{\gamma} = A(c_n,t) + A'(c_n,t-1)\left(1-\frac{1}{\gamma}\right) \tag{11}$$

Once the temporal smoothing has been performed, the winning concept c for each time frame is chosen by selecting the one with the highest activation level. For example, for speech recognition a median filter of 250 ms and $\gamma=6$ 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 or lip movement features for speech). This transforms frequency and activation matrices in the form $\mathbf{F}_{k,w}(a_i,a_j|c)$ and $\mathbf{Q}_{k,w}(a_j|a_i,c)$, where w denotes 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_n at time t of

$$A(c_n,t) = \sum_{w=1}^{|w|}\left[\sum_{m=1}^{K} \mathbf{Q}_{k_m,w}(X_t|X_{t-k_m},c_n)\right]*\omega_w \tag{12}$$

where ω_w 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 never bias or degrade the recognition process. Also, if $N_{D,c} \gg N_A^2$, where $N_{D,c}$ is the number of transitions in the training data for context c, is satisfied for all $\mathbf{c}$, it is not necessary to use ω_w at all due to automatic neglecting of uninformative transitions (unpublished results). However, experiments with multiple stream pattern discovery are out of the scope of this paper and will be addressed in future work.

3.4. The baseline system: n-grams

In order to demonstrate the utility of modeling long-range dependencies with mixtures of bi-gram statistics, the proposed system was compared against a standard n-gram based classifier. n-Grams were chosen as a reference because they represent the methodological starting point from which the CM departs in order to overcome the Markov assumption and the claimed sparsity problems of the n-grams (see discussion in Section 2). In the n-gram classifier, the frequencies $f(g^n|c_i)$ of all n-tuples of order $n=1$–5 were computed for all sequences X occurring in context c_i, i.e. element frequencies were also used. Higher order ($n>5$) n-tuples were excluded from the analysis due to sparseness of the data, since nearly all of them occurred only once or twice in the entire data, and also because the amount of unique n-tuples was too high to fit into the 12 GB memory of the workstation. The probability distributions P' of order n n-tuples during each context c were then computed by normalizing the frequency distributions to sum up to one. Then the probability that a n-tuple $g^n=\{a_1, a_2, \ldots, a_n\}$ occurs in context c_i was computed by having:

$$P(\mathbf{g}^n|c_i) = \frac{P'(\mathbf{g}^n|c_i)}{\sum_{j=1}^{N_C} P'(\mathbf{g}^n|c_j)} \tag{13}$$

During recognition, the instantaneous activation of model c_i at time t was computed according to Eq. (14), where $\mathbf{g}_t^n$ is the n-tuple $\mathbf{g}$ of order n starting at time t, and N is the maximum n-tuple order:

$$A(c_i,t) = \sum_{n=1}^{N} P(\mathbf{g}_t^n|c_i) \tag{14}$$

Similarly to the CM, the activation of the models was then computed in parallel for all C, smoothed temporally using the same parameters as in the CM, and then the model with the largest activation value was chosen as the recognition hypothesis for each moment of time. In general, the baseline system was similar to the CM except that the basic elements in the analysis were n-tuples up to the order $n=5$ instead of mixtures of bigrams at varying temporal distances.

In addition to comparing the performance of the CM to the n-gram recognizer, the normalization procedures of the CM (Eqs. (7) and (8)) were also compared to an alternative normalization in order to further justify the chosen methodology. Instead of computing the stochastic matrices according to Eq. (7), it is possible to model structure of sequences by computing the joint probabilities of element pairs at distances k in the context of c

$$\widehat{\mathbf{P}}_k(a_i,a_j|c) = \frac{\mathbf{F}_k(a_i,a_j|c)}{\sum_{x=1}^{N_A}\sum_{y=1}^{N_A} \mathbf{F}_k(a_x,a_y|c)} \tag{15}$$

and then normalizing these probabilities according to Eq. (8). The use of joint probability means that all transitions between sequence elements take place in a global probability space. In practice this leads to a situation where the most dominant trajectories accumulate large portion of the probability mass, whereas rarely occurring variants of patterns receive very small probabilities. On the contrary, the use of stochastic matrices in Eq. (7) allows the existence of several parallel paths that can be equally likely for a given model, since the probability mass is conditioned locally for each sequence element that acts a starting point for the transition. The hypothesis was that this additional degree of freedom enhances pattern discovery performance from the joint probabilities. We also compared the performance with and without application of the normalization across all models in Eq. (8).

4. Experiments

4.1. Experiment I: keyword discovery

The first experiment consists of a word learning task that simulates the problem that human infants face when they are attempting to learn their first word-referent associations. The purpose of the experiment is to demonstrate how the learning algorithm extracts statistically significant structures from the VQ speech in relation to contextual information that is aligned at the utterance level, but not on the word level. The experiment can be considered as an instance of cross-situational learning [4], where the co-occurrence statistics of two modalities need to be processed in an efficient manner in order to find the proper dependencies between the two. In addition, the usefulness of the inclusion of information from several lags is also demonstrated using this data set.

4.1.1. Material and evaluation

The speech material consisted of the CAREGIVER corpus [35] that contains continuously spoken child directed speech in English. In the Y2-version of the corpus, each utterance contains one to four of 50 possible keywords and a surrounding carrier sentence (mean = 2.94 keywords per sentence). The carrier sentences vary across utterances, e.g. "*Smiling* ***daddy has*** *the* ***fish***", "*He* ***gives*** *a* ***small cookie***", "*Do you* ***like*** *a* ***happy cat***", and "***Mummy looks*** *at a* ***big tree***" (keywords emphasized). Each utterance is paired with an unordered set of contextual tags that indicate which objects are present in the surrounding visual scene that the learner is paying attention to, i.e., the algorithm has to associate highly variable acoustic word forms and the contextual variables together. The training set consisted of 4×2000 utterances spoken by four different speakers (two males, two females) and the test set consisted of 4×396 novel utterances spoken by the same speakers, yielding a test set of a total of 1584 utterances. The total duration of the audio is 7 h and 20 min, including silence.

The evaluation was performed by having the algorithm to provide an ordered set of N word hypotheses for each utterance, where N was the true number of target words in the 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 a priori knowledge about word string length has been shown to increase the HMM recognition rate by approximately 2% in a similar task with noise [36].

4.1.2. Data preparation

In order to obtain a discrete sequence representation for speech, Mel-frequency cepstral coefficients (MFCCs; 12 first coefficients) were extracted from the audio with a 32 ms Hamming window and 10 ms frame shifts. In addition, the log energy of the signal was included to help differentiation between speech and silence, since no separate voice activity detection was used. The log energy and spectral tilt (the first coefficient) were dampened by a factor of 0.3 to reduce their relative weight in quantization. VQ codebooks of size $N_A = 64$ (single speaker) and $N_A = 128$ indices (four speakers) were created by k-means clustering with a subset of 15,000 MFCC vectors from the training set, and then all utterances were converted into VQ sequences with 10 ms frames. The CM matrices were trained with speech VQ data and the contextual tags using 15 lags $\boldsymbol{k} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 17, 23\}$ (notice that lags do not have to be consecutive as in Mixed-order Markov models). In recognition, activation curves were integrated with $\gamma = 6$ frames and a median filter of 25 frames (250 ms) in length.

4.1.3. Results for keyword recognition

Fig. 3 displays the keyword recognition rates for all four speakers (left) and a single female speaker (right) as a function of the number of trained utterances. In addition to the standard CM (*stochastic + normalization*) and baseline n-gram recognizer, also the joint probability approach in Eq. (15) (*joint + normalization*) and CM without the normalization of Eq. (8) (*stochastic*) are shown. Joint probability performance without the normalization in Eq. (8) is not shown since it did not receive notably above chance performance at any stage.

As can be observed, the recognition rate of the CM starts to increase gradually directly from the beginning and after training the full train set of 8000 utterances the mean recognition rate achieves an accuracy of 92.41% correct detections. In the right panel of Fig. 3 one can see the same learning process, but now using speech from only one speaker for training and testing. In this case the learning is approximately four times faster as in the case of four speakers. This difference in learning curves using data from one and four speakers mainly demonstrates the intrinsic variability in speech. Since realizations of the same word from different speakers have very different acoustic representations, their mappings to a discrete VQ space are also very different. This makes generalization across speakers difficult, and most of the words have to be learned separately for each speaker, thereby multiplying the amount of required training data by a large factor. Despite this, it is important to note that the overall distinctiveness of the models does not degrade from single speaker condition even though there now exists many parallel speaker-specific trajectories for each c in the multiple speaker condition.

Performance of the standard n-gram recognizer is poor in the task (Fig. 3). For both multiple and single speaker conditions, the overall keyword recognition performance with n-grams stays below 40% correct detections when all n-grams from the training data are used as a model for each context (a total of 1,029,192 unique n-grams for 4 speaker data). If the number of parameters is limited to N most frequently occurring n-grams per each model, the performance decreases monotonically as the value of N is decreased. Also, if the maximum order of the n-grams is decreased, or any subset of the orders $n = \{1, \ldots, 5\}$ is used, the recognition rate becomes worse. Moreover, the computational complexity and memory requirements of the n-gram system are

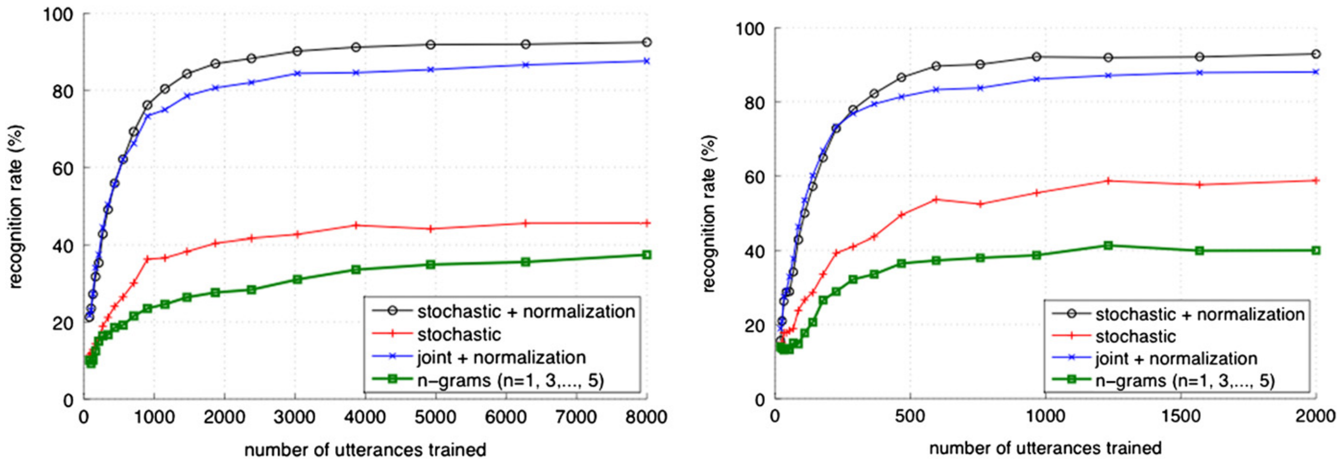

Fig. 3. Recognition results for full four speaker training and test set (left) and for a single female speaker (right). Recognition rates are shown using the three different probability normalization techniques: normal CM (stochastic + normalization), transition matrices without normalization across concepts (stochastic), and joint distributions instead of transition matrices (joint + normalization). Also, the baseline result with n-grams are shown.

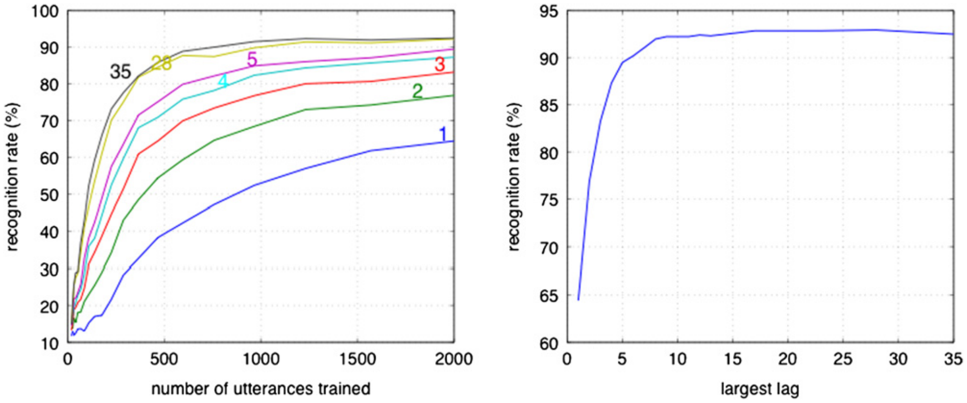

Fig. 4. *Left*: recognition learning curves for different lag values in the single speaker condition. The number on the curve indicates the largest lag included in the analysis (e.g., 5 means lags $\boldsymbol{k}=\{1, 2, 3, 4, 5\}$). *Right*: keyword recognition rates after full training as a function of number of lags.

huge when compared to the CM. As an example, training of 2000 utterances with CM using seventeen lags takes 34 s and recognition of one utterance takes approximately 37 ms, whereas the training time of the n-grams is 74 min and recognition of one utterance takes approximately 3.3 s (MATLAB environment[4] with 4 × 3.2 GHz Intel Xeon, 12 GB RAM; codebook of size $N_A=64$). In general, the result confirms that the speech signals are too complex to be directly modeled with rigid n-gram-based approaches (being one reason for the existence of HMM speech recognizers in the first place).

The hypothesis that models based on joint probability of sequence elements (Eq. (15)) perform worse than transition probabilities can be also verified from Fig. 3: the joint probabilities systematically fall behind the standard CM approach, although the difference is not very large in the early stages of the learning. One can also see that the normalization across models (Eq. (8)) is an important discriminative step in the training, and without it the context-specific stochastic transition matrices at different lags perform relatively poorly in recognition. Interestingly, the element joint distributions without normalization across models perform only at a chance level. This is mostly explained by the fact that the probability values of all transitions in joint distribution models are highly affected by the overall frequencies of different elements. In the case of speech, silence and closures are modeled by a small number of VQ indices that obtain very high frequencies (transitions from silence to silence). When normalization (Eq. (8)) is applied, the effect of overall frequency disappears since silence is present in all models in a more or less equal manner.

Fig. 4 displays the keyword classification performance in the single speaker condition when the number of lags is varied. Left panel in the figure shows the performance for seven different lag parameters, where the number on each curve indicates the largest lag included in the analysis. Right panel shows the performance after full training for lags from $\boldsymbol{k}=\{1\}$ up to $\boldsymbol{k}=\{1,2, \ldots, 35\}$. The results show that the recognition rate increases when the modeling distance is increased up to approximately 15–20 frames (150–200 ms). This confirms that there are important structures in speech distributed at the long-range dependencies that are not captured with sequential analysis of local dependencies. Also, the performance at lags $\boldsymbol{k}=\{1,2, \ldots, 5\}$ is already nearly 90% correct recognitions, whereas n-grams of order $n=\{1, \ldots, 5\}$ achieve a recognition rate of only 40%. As a useful property of the algorithm, the performance does not drop significantly even though the lag values are increased beyond the point of optimal performance.

As an interesting detail, after linear scaling, the early part of the recognition rate curve is a very close match to the inverse of the MIF curve in Fig. 1 up to approximately 20 frames, i.e. to the point after which the recognition rate stops increasing.

In general, the experiment shows that the CM is able to model structural properties of the signal that are relevant for the given contextual information even when there is a large amount of overlap between the training samples from different context categories (keywords cover only 10–15% of time in each utterance) and large variation between realizations of the patterns (inherent property of speech).

4.2. *Experiment II: spoken digit recognition in noise*

In the second experiment, the task of the learning algorithm is to learn and recognize words corresponding to the English digits (0–9, "*oh*" and "*zero*") from continuous speech that contains varying levels of additive noise. Data from TIDIGITS corpus was chosen for this experiment since it has been widely used in the speech recognition research, making comparison to supervised speech recognizers possible.

4.2.1. *Material and evaluation*

The material used in the experiment consisted of the TIDIGITS corpus [37] that contains continuously spoken digit sequences (one to seven digits per utterance) in different dialects of American English from a total of 225 different speakers (111 males, 114 women, $f_s=16$ kHz). The training material consisted of the full male and female training set ($N=8623$ utterances). The test data consisted of the full adult test set ($N=8700$ utterances). However, since the lack of speech specific 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 the same digit (like "six-six-nine-two") were excluded from the test set. Otherwise the evaluation procedure was identical to the previous experiment.

4.2.2. *Data preparation*

The data was prepared in a similar fashion to experiment I using a 32 ms Hamming window with 10 ms shifts to extract 12 MFCC coefficients, with the first coefficient (spectral tilt) dampened by a factor of 0.3. The only differences were that the log-energy was not used in this experiment as a feature, and the utterance-based cepstral mean and variance normalization (CMVN; [38]) was applied to the MFCC vectors in order to

[4] It has to be taken into consideration that despite relatively optimized code, the MATLAB is not an optimal platform for processing of n-grams due to the absence of effective hashing routines.

increase robustness of the vector quantization in noise. A codebook of size $N_A=150$ was used. Context tags related to each utterance were extracted directly from the signal annotation, one for each digit including "*oh*" and "*zero*", yielding a total of $N_C=11$ different tags. As an outcome, each utterance was described as one VQ sequence and an unordered set of tags related to the digit words in the utterance. Activation curves were integrated with $\gamma=6$ frames and a median filter 250 ms in length. The same 15 lags as in the experiment I were used.

4.2.3. Results

Clean speech word recognition accuracy in the digit recognition task was 95.24% for the TIDIGITS test set with a codebook of size $N_A=150$. A larger codebook of 300 indices was also tested, increasing the recognition rate to 96.11%. However, the larger codebook did not yield notable gain in noise experiments, but actually makes the VQ more susceptible to noise, and was therefore not used in further experiments.

An example of the recognition process is shown in Fig. 5, where the utterance "three-four-one-two-six" is being analyzed. Activation curves after the cumulative activation of Eq. (10) are shown on the left. The right panel shows the same signal after median filtering and inhibition of non-winning concepts, now each recognizer shown on its own row.

As the left graph in Fig. 5 illustrates, 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 the activations of non-related parts are below zero. Additionally, the change points between concepts (Fig. 5, right) provide an accurate word and/or morpheme segmentation of the input. This was noted by a manual inspection using English and Finnish continuous speech. The only incorrect point in the segmentation shown in Fig. 5 is the ending of the utterance, since it consists of silence that is not modeled separately. This means that there are no competing models that would inhibit the last digit "*six*", leading to very slow decay of activation.

Noise robustness of the algorithm was tested using white Gaussian noise (WGN) and non-stationary factory, babble, and car (Volvo) noises taken from the NOISEX database (http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html). Training of the codebook and CM were performed using clean speech.

Fig. 6 displays the results as a function of SNR. The CM algorithm performs relatively well at a SNR of 20 dB in all cases (92.77%, 89.74%, 85.99% and 85.83% for car, factor, babble and white noise, respectively), but the recognition rate starts to degrade considerably for all but car noise below that level.

The clean speech result is below state-of-the-art HMM-based speech recognizers using continuous distributions and full covariance matrices, since they obtain nearly perfect recognition for clean speech. However the recognition rates at low SNRs are still very comparable to the results reported with continuous density HMMs, with and without noise compensation [39–41]. For example, Cooke et al. [39] report approximately 71% and 11% words correct for factory noise of 10 and 0 dB SNR, respectively, using cepstral mean normalization. For the discrete output HMMs and clean speech, a result of 96.1% string recognition rate and 98.6% word recognition have been reported in the same task [42]. For the Non-Negative Matrix Factorization (NMF) also using statistical models of VQ indices collected at a number of lags, word recognition rate of 92.6% has been reported [43] without a priori knowledge of the number of words.

The gender-dependent CM models were also tested, but they did not lead to a significant increase in accuracy when compared to modeling of both genders simultaneously. However, it was beneficial to include training samples from both males and females in the creation of the codebook, yielding a small (~1–2%)

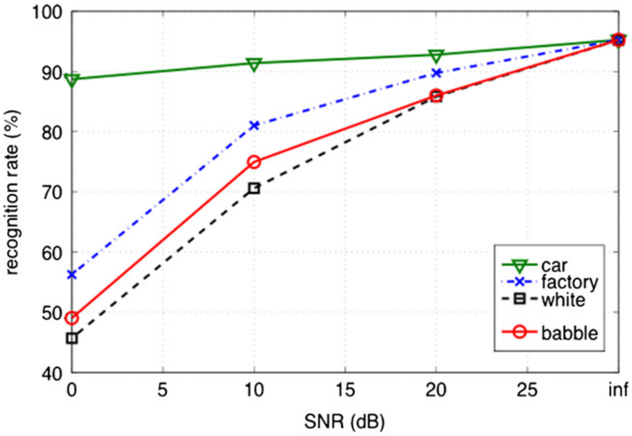

Fig. 6. Digit recognition in car, factory, white, and babble noise.

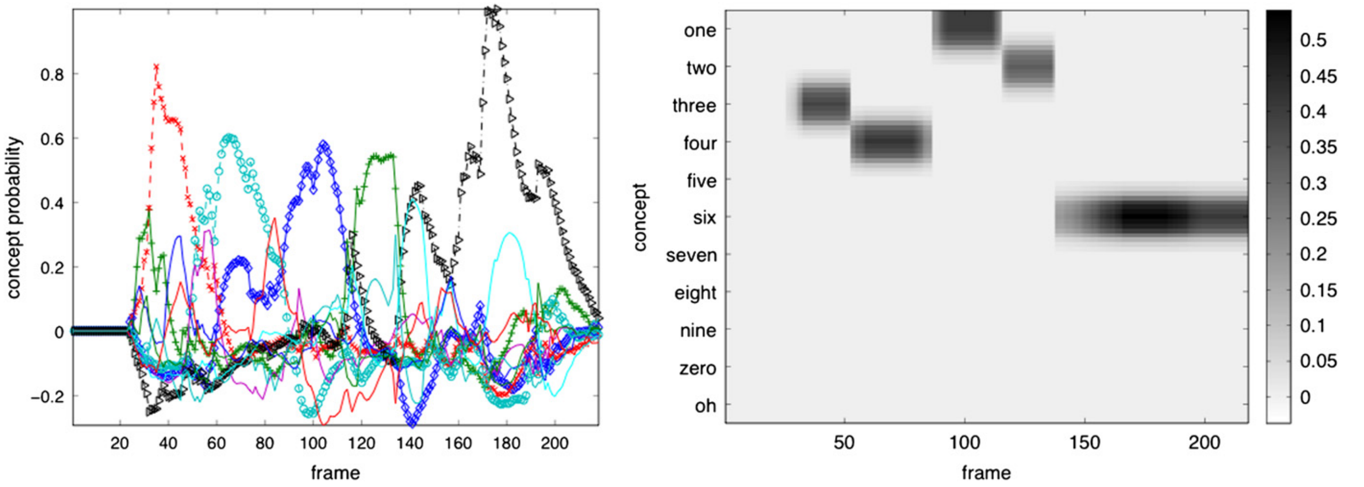

Fig. 5. *Left*: cumulative activation curves of all 11 recognizers in recognition of the utterance "three-four-one-two-six". *Right*: activation after median filtering and inhibition.

but still significant impact on recognition accuracies, depending on the noise type and level.

It is important to note that the results of the CM and the HMM (and also Artificial Neural Network—ANN)-based speech recognizers should not be compared directly, since a standard recognizers utilize iterative training and expert knowledge in the task in terms of number of states, order of words in utterances, or even temporal alignment of training samples. On the other hand, CM only assumes the time scale in which interesting events occur (lags) and annotation of contextual variables of interest at a level where the systematical relationship between annotation and the signal contents is above chance level. However, in practice, the amount of manual work required for training of the HMMs is often decreased by bootstrapping the system with a number of manually segmented training samples and then using the recognizer to segment the remainder of the training data for the final training (e.g., [42]; see also [44] for ANNs). For material like TIDIGITS, it is also possible to bootstrap HMM without annotated training material using the forward–backward algorithm. However, without the knowledge of the proper number of states per model, it is very difficult to end up with high quality HMM models. The CM, on the other hand, learns the relevant structures for each model using co-occurrence statistics of time-series input and context input, and no temporal alignment or ordering between the training data and training labels is required at all. Since the CM algorithm provides word level segmentation of the data after learning, it can also be used for fast and automatic bootstrapping of HMM recognizers.

It should also be noted that all processing in training of the CM takes place during the first pass, making it computationally extremely feasible. On the other hand, the word recognition task here was simplified such that the number of target words was always known. It is also noteworthy that the CM does not utilize first- or second-order derivatives of the MFCC features, reducing the size of the feature significantly from normal HMM-based speech recognizer. The use of delta and delta–delta information has only a very minor improving effect on the overall word recognition performance, as long as the overall number of lags and the frame rate are sufficiently high.

5. Conclusions and future work

It was shown that given a set of data sequences as an input and a set of informative context tags for each sequence, the CM algorithm is able to create structural models that associate the presence of a specific contextual variable with relevant co-occurring patterns in the input sequence. This model can be used for the recognition of similar patterns in future input and can handle large amounts of variability and noise in the sequences when compared to standard Markov chains. This is because the CM does not make the Markov property assumption, i.e. it does not attempt to pack all the information of 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. The CM algorithm is 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.

From the perspective of weakly supervised word learning from continuous speech, the CM is able to learn statistical models for separate words and recognize them with high accuracy without segmentation of the training data to proper training samples for each word. Noise robustness of the CM is also relatively good when compared to the supervised HMM algorithms [39–41], although it does not employ any kind of special mechanism for dealing with noisy input in addition to cepstral mean and variance normalization. However, since the CM in its basic form lacks a mechanism for detecting and decoding speech specific word-like units, the recognition task was slightly simplified. The use of a decoding mechanism tailored especially for word or phone recognition would bring the system closer to real speech recognition applications, where the number of words cannot be assumed beforehand, and would probably actually help in speech recognition by including speech and language specific constraints to the processing.

In addition to speech, the CM method has been applied to a purely supervised pattern recognition task, namely audio environment classification. By using the experimental setup of Ma et al. [45], an environment classification accuracy of 96% was obtained with the CM (unpublished results), whereas Ma et al. [45] report classification accuracy of 92% for a continuous density HMM-based recognizer. Also, the research group of the authors is currently studying the applicability of the CM algorithm to automatic discovery and classification of events in pre-term infant EEG-signals.

The algorithm is efficient in terms of computational complexity. The complexity of the normalizations in training is $O(N_C K N_\mathbf{A}^2)$, where N_C is the number of models, K is the number of lags and N_A is the size of the alphabet. In batch training, these normalizations have to be performed only once. Recognition (without temporal smoothing) requires only retrieval and summing of $N_C KT$ values from memory, where T is length of the sequence. The computational swiftness makes the CM algorithm suitable for, e.g., portable devices where CPU usage and power consumption have to be low. The maximum memory requirement for storing all models is $KN_\mathbf{A}^2 N_C B$ bits, where B is the number of bits allocated for each parameter. These memory requirements can be further relaxed to a large degree by exploiting the sparseness of transition statistics. As an example, the CM algorithm has already been implemented to standard smartphones for real-time learning and recognition of user activities (to be published). The use of discrete input data makes the approach also suitable for low bit-rate distributed recognition, where the feature extraction and actual processing and data storage are performed at different sites (see, e.g., [46]).

Finally, although the CM algorithm presented in this paper works in a discrete space, it can be extended to a semi-continuous domain similarly to the HMMs [18]. If the VQ preprocessing is replaced by Gaussian mixture modeling (GMM) of the input features, the input stream will consist of the indices of the N best Gaussians and their corresponding posterior probabilities. Although this increases the computational complexity of the algorithm, it enables the modeling of signal trajectories with high accuracy without increasing the training data requirements. By performing training and recognition using the GMM input, it should be possible to adapt the GMM parameters for new input properties and/or increase discrimination power in the recognition task with the help of feedback from the CM level (cf., [18]). These issues along totally unsupervised learning and extension to fully multimodal time-series analysis will be the central topics in future research of the CM.

Acknowledgements

This research was funded as part of the EU FP6 FET project Acquisition of Communication and Recognition Skills (ACORNS), Contract no. FP6-034362, and as a part of the Towards Multimodal Pattern Discovery (TOMU) -project funded by Nokia NRC Tampere. The authors would also like to thank the two anonymous reviewers for their useful comments on the manuscript.

References

[1] D. Crandall, D. Huttenlocher, Weakly supervised learning of part-based spatial models for visual object recognition, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 16–29.
[2] R. Fergus, P. Perona, A. Zisserman, Weakly supervised scale-invariant learning of models for visual recognition, International Journal of Computer Vision 71 (3) (2007) 273–303.
[3] M. Vasconcelos, G. Carneiro, N. Vasconcelos, Weakly-supervised top-down image segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 1001–1006.
[4] L. Smith, C. Yu, Infants rapidly learn word-referent mappings via cross-situational statistics, Cognition 106 (3) (2008) 1558–1568.
[5] K. Gold, M. Doniec, C. Crick, B. Scasselati, Robotic vocabulary building using extension inference and implicit contrast, Artificial Intelligence 173 (2009) 145–166.
[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the Society for Information Science 41 (6) (1990) 391–407.
[7] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, K. Weinberger, Supervised semantic indexing, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, 2009, pp. 187–196.
[8] U.K. Laine, O. Räsänen, T. Altosaar, J. Driesen, G. Aimetti, G. Henter, Methods for enhanced pattern discovery in speech processing, EU FP6 FET ACORNS project deliverable, 2008, 〈http://lands.let.ru.nl/acorns/documents/Deliverables_Y2/Del%202.2.pdf〉.
[9] L. Boves, L. ten Bosch, R. Moore, ACORNS—towards computational modeling of communication and recognition skills, in: Proceedings of the IEEE International Conference on Cognitive Informatics, 2007, pp. 349–356.
[10] W. Liao, Clustering of time series data—a survey, Pattern Recognition 38 (2005) 1857–1874.
[11] W. Li, Mutual Information Functions of Natural Language Texts, Santa Fe Institute preprint SFI-89-008 (1989).
[12] W. Li, Mutual information functions versus correlation functions, Journal of Statistical Physics 60 (5/6) (2000) 823–837.
[13] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.
[14] F. Samaria, S. Young, HMM-based architecture for face identification, Image and Vision Computing 12 (8) (1994) 537–543.
[15] O.E. Agazzi, S-S. Kuo, Hidden Markov model based optical character recognition in the presence of deterministic transformations, Pattern Recognition 26 (12) (1993) 1813–1826.
[16] T. Plötz, G.A. Fink, Pattern recognition methods for advanced stochastic protein sequence analysis using HMMs, Pattern Recognition 39 (2006) 2267–2280.
[17] L.R. Rabiner, B.H. Juang, An introduction to hidden Markov models, IEEE ASSP Magazine 3 (1) (1986) 4–16.
[18] X.D. Huang, M.A. Jack, Unified techniques for vector quantization and hidden Markov modeling using semi-continuous models, in: Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP'89), 1989, pp. 639–642.
[19] M. Gales, S. Young, The application of hidden Markov models in speech recognition, Foundations and Trends in Signal Processing 1 (3) (2008) 195–304.
[20] J. Chen, Y. Huan, Q. Li, K. Paliwal, Recognition of noisy speech using dynamic spectral subband centroids, IEEE Signal Processing Letters 11 (2) (2004) 258–261.
[21] H.A. Engelbrecht, J.A. du Preez, Efficient backward decoding of high-order hidden Markov models, Pattern Recognition 43 (2010) 99–112.
[22] M. Casar, J.A.R. Fonollosa, A n-gram approach to overcome time and parameter independence assumptions of HMM for speech recognition, in: Proceedings of the ISCA European Signal Processing Conference (EU-SIPCO), 2007.
[23] T. Kobayashi, J. Furuyama, K. Masumitsu, Partly hidden Markov model and its application to speech recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999, pp. 121–124.
[24] L-M. Lee, J-C. Lee, A study on high-order hidden Markov models and applications to speech recognition, Lecture Notes in Computer Science/ Advances in Applied Artificial Intelligence 4031 (2006) 682–690.
[25] A.E. Raftery, A model for high-order Markov chains, Journal of Royal Statistical Society B 47 (3) (1985) 528–539.
[26] A. Berchtold, A.E. Raftery, The mixture transition distribution model for high-order Markov chains and non-Gaussian time series, Statistical Science 17 (3) (2002) 328–356.
[27] W.K. Ching, E.S. Fung, M.K. Ng, High-order Markov chain models for categorical data sequences, Naval Research Logistic 51 (4) (2004) 557–574.
[28] A.E. Raftery, A new model for discrete-valued time series: autocorrelations and extensions, Rassegna di Metodi Statistici ed Applicazioni 3–4 (1985) 149–162.
[29] A. Berchtold, Modélisation autoregressive des chaînes de Markov: utilization dune matrice différente pour chaque retard, Revue de Statistique Appliquée 3 (1996) 5–25.
[30] A. Berchtold, Estimation in the mixture transition distribution model, Journal of Time Series Analysis 22 (4) (2001) 379–397.
[31] L. Saul, F. Pereira, Aggregate and mixed-order Markov models for statistical language processing, in: Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, Rhode Island, USA, 1997, pp. 81–89.
[32] Y. Singer, Adaptive mixtures of probabilistic transducers, in: D. Touretzky, M. Mozer, M. Hasselmo (Eds.), Advances in Neural Information Processing Systems, 8, MIT Press, Cambridge, MA, 1996, pp. 381–387.
[33] K. Smith, A. Smith, R. Blythe, P. Vogt, Cross-situational learning: a mathematical approach, in: P. Vogt, Y. Sugita, E. Tuci, C. Nehaniv (Eds.), Symbol Grounding and Beyond, Springer, Berlin, 2006, pp. 31–44.
[34] O. Räsänen, U.K. Laine, T. Altosaar, Computational language acquisition by statistical bottom-up processing, in: Proceedings of the Interspeech '08, 2008, pp. 1980–1983.
[35] T. Altosaar, L. ten Bosch, G. Aimetti, C. Koniaris, K. Demuynck, H. van den Heuvel, A speech corpus for modeling language acquisition: CAREGIVER, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC), Malta, 2010, pp. 1062–1068.
[36] J. Kim, R. Haimi-Cohen, F. Soong, Hidden Markov models with divergence based vector quantized variances, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99), 1999, pp. 125–128.
[37] R.G. Leonard, A database for speaker-independent digit recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'84), vol. 3, 1984, pp. 328–331.
[38] O. Viikki, K. Laurila, Cepstral domain segmental feature vector normalization for noise robust speech recognition, Speech Communication 25 (1) (1998) 133–147.
[39] M. Cooke, P. Green, L. Josifovski, A. Vizinho, Robust automatic speech recognition with missing and unreliable acoustic data, Speech Communication 34 (2001) 267–285.
[40] P. Renevey, Speech Recognition in Noisy Conditions Using Missing Feature Approach, Ph.D. Thesis, Lausanne, EPFL, 2000.
[41] B. Raj, R.M. Stern, Missing-feature approaches in speech recognition, IEEE Signal Processing Magazine 22 (2005) 101–116.
[42] R. Cardin, Y. Normandin, R. De Mori, High performance connected digit recognition using maximum mutual information estimation, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1991, pp. 533–536.
[43] H. Van hamme, HAC-models: a novel approach to continuous speech recognition, in: Proceedings of the Interspeech '08, 2008, pp. 2554–2557.
[44] K.P. Unnikrishnan, J.J. Hopfield, D.W. Tank, Speaker-independent digit recognition using a neural network with time-delayed connections, Neural Computation 4 (1992) 108–119.
[45] L. Ma, B. Milner, D. Smith, Acoustic environment classification, ACM Transactions on Speech and Language Processing 3 (2) (2006) 1–22.
[46] S. So, Efficient Block Quantisation for Image and Speech Coding, Doctoral Thesis, Griffith University, Australia, 2005.

Okko Räsänen received his M.Sc. (Tech.) degree in Language Technology from the Helsinki University of Technology (TKK), Finland, in 2007, where he has been working on his Ph.D. degree in Language Technology since. His research interests include unsupervised and weakly supervised learning, cognitive aspects of language acquisition and processing, and basic research in speech processing.

Unto K. Laine received his M.Sc. degree from Tampere University of Technology (1972) and Ph.D. (Eng.) from Helsinki University of Technology (TKK; 1989). He has held the chair of speech technology at TKK since 2002 (from 1.1.2010 on, Helsinki University of Technology (TKK) has become Aalto University School of Science and Technology).