MODELING SPEECH WITH SUM-PRODUCT NETWORKS: APPLICATION TO BANDWIDTH EXTENSION

Robert Peharz, Georg Kapeller, Pejman Mowlaee and Franz Pernkopf

Signal Processing and Speech Communication Lab
Graz University of Technology

ABSTRACT

Sum-product networks (SPNs) are a recently proposed type of probabilistic graphical models allowing complex variable interactions while still granting efficient inference. In this paper we demonstrate the suitability of SPNs for modeling log-spectra of speech signals using the application of artificial bandwidth extension, i.e. artificially replacing the high-frequency content which is lost in telephone signals. We use SPNs as observation models in hidden Markov models (HMMs), which model the temporal evolution of log short-time spectra. Missing frequency bins are replaced by the SPNs using most-probable-explanation inference, where the state-dependent reconstructions are weighted with the HMM state posterior. According to subjective listening and objective evaluation, our system consistently and significantly improves the state of the art.

Index Terms— graphical models, SPN, HMM, speech bandwidth extension

1. INTRODUCTION

Probabilistic graphical models (PGMs) [1, 2] enjoy great popularity in the speech and signal processing communities. As an example, hidden Markov models (HMMs) [3] are one of the most popular probabilistic models for modeling sequential data, with a vast amount of applications, such as speech recognition/synthesis, natural language processing and bio-informatics. PGMs aim to trade-off computational requirements of probabilistic inference and the amount of statistical independence assumptions. However, while most research in PGMs focuses on novel techniques for learning and inference, application driven research usually restricts to more "simplistic" models, like naive Bayes classifiers, HMMs, Gaussian mixture models (GMMs), Markov random fields restricted to pair-wise interactions, etc. The reason for this is that inference in these models is conceptually simple and computationally tractable. The simplicity of these models, however, sacrifices model-expressiveness and possibly performance of the incorporating system.

In [4, 5, 6] and related work, novel types of probabilistic models emerged which allow to control the inference cost during learning but still modeling complex variable dependencies. Using the *differential approach* introduced in [4], inference is also conceptional easy in these models. In this paper, we consider sum-product networks (SPNs) introduced in [6]. SPNs can be interpreted as Bayesian networks with a deep hierarchical structure of latent variables with a high degree of context-specific independence. In this way, SPNs can model highly complex variable interactions with little or no conditional independencies among the model variables. Furthermore, SPNs can be interpreted as a neural network representing an inference machine, where inference is *linear* in the networks size, i.e. in the number of nodes and edges in the network. To the best of our knowledge, we describe the first application of SPNs to a speech related task, namely artificial bandwidth extension (ABE) of lowpass-filtered (telephone) speech. Motivated by the success of SPNs on the task of image completion [6], we use SPNs to complete the high frequency parts of log-spectrograms, lost due to the telephone bandpass filter. Specifically, we use SPNs as observation models in HMMs modeling the temporal evolution of the log-spectrum. To infer the marginal HMM state distributions we use the forward-backward algorithm, where missing frequency bins are marginalized by the SPN models. The high frequency bins are reconstructed by most-probable-explanation inference [6], where the reconstructions of the state-dependent SPNs are weighted by the state posterior. The resulting log-spectrograms exhibit speech structures similar to the original wide-band speech, and the resynthesized speech signals clearly exhibit an improved speech quality due to the added high frequency content. Using log-spectral distortion as objective measure, we report consistent and significant improvement over state-of-the-art methods.

The paper is organized as follows: In section 2 we review SPNs. In section 3 we describe our approach for ABE using SPNs embedded in an HMM. In section 4 we discusses resynthesis of time signals from bandwidth extended log-spectrograms. In section 5 we present our experiments and section 6 concludes the paper.

2. SUM-PRODUCT NETWORKS

Let $X_m, m \in \{1, \dots M\}$ denote random variables and let x_m be an instantiation of X_m. We define $\mathbf{X} := \{X_1, \dots, X_M\}$ and $\mathbf{x} := \{x_1, \dots, x_M\}$, and for any index set $I \subseteq \{1, \dots, M\}$ we define $\mathbf{X}_I := \{X_m : m \in I\}$ and $\mathbf{x}_I := \{x_m : m \in I\}$.

An SPN is an acyclic directed graph whose internal nodes are sum and product nodes. Each internal node recursively calculates its value from the values of its child nodes: sum nodes calculate a *non-negatively weighted* sum of the values of their child nodes, where the non-negative weights are associated with the emanating edges of the sum node. Product nodes calculate the product of their child nodes' values. While SPNs generally can have multiple roots [7], in this paper we assume SPNs with a single root. The value of the root node is the output of the SPN, while the input of the SPN is provided by its leave nodes. In [6], the leaves of an SPN were defined to be *indicator nodes* of discrete random variables, such that the SPN represents the *network polynomial* of a Bayesian network [4]. In [8, 7, 9] the concept of SPN leaves was generalized such that they represent *tractable* distributions over single variables, or (small) sets of variables. More precisely, when N is a leave of an SPN, then

This work was supported by the Austrian Science Fund (project number P25244-N15). The work of P. Mowlaee was supported by the European project DIRHA (FP7-ICT-2011-7-288121) and K-Project ASD.

the value of N for some input $\mathbf{x}$ is $N(\mathbf{x}) := p_N(\mathbf{x}_{\mathbf{sc}(N)})$, where the *scope* $\mathbf{sc}(N) \subseteq \{1, \ldots, m\}$ are the indices of variables associated with N, and p_N is a tractable distribution over $\mathbf{X}_{\mathbf{sc}(N)}$. p_N can either be a probability mass function (PMF) or a probability density function (PDF). Generally, there are several leave nodes with the same scope, representing a collection of distributions over the same variables. This view of SPN leaves subsumes the definition using indicator nodes in [6], since an indicator function is a special case of a PMF, assigning all probability mass to a single state.

Concerning some internal node N, i.e. a sum or a product node, we define $\mathbf{sc}(N) := \bigcup_{C \in \mathbf{ch}(N)} \mathbf{sc}(C)$, where $\mathbf{ch}(N)$ denotes the children of N. Let R denote the root node of the SPN, and assume w.l.o.g. that $\mathbf{sc}(R) = \{1, \ldots, M\}$. Then an SPN defines a probability distribution over $\mathbf{X}$ as $p_{SPN}(\mathbf{x}) \propto R(\mathbf{x})$, i.e. by its normalized output. In order to perform efficient inference (e.g. marginalization, most-probable explanation, conditional marginals), an SPN should be *valid* [6]. A sufficient condition for validity is when the SPN is *complete* and *decomposable*, defined as follows [6]:

- *Completeness*: For any two children C, C' of any sum node, it must hold that $\mathbf{sc}(C) = \mathbf{sc}(C')$.
- *Decomposability*: For any two children C, C' of any product node, it must hold that $\mathbf{sc}(C) \cap \mathbf{sc}(C') = \emptyset$.

When an SPN is complete and decomposable, and when the non-negative weights are normalized to 1 for each sum node, then the output is already normalized and $p_{SPN}(\mathbf{x}) = R(\mathbf{x})$. A complete and decomposable SPN can be naturally interpreted as a recursively defined distribution: product nodes serve as *cross-overs* of distributions with non-overlapping scope, representing a local independence assumption; sum nodes represent *mixtures* of distributions, dissolving these independence assumptions [8, 7]. Since sum nodes represent mixtures, one can associate a latent random variable with each sum, which opens the door for expectation-maximization algorithms [6].

In [6], an algorithm was proposed for learning SPNs on data organized as a rectangular array (e.g. images). Starting with the whole rectangle (the root), the algorithm recursively performs all decompositions into two sub-rectangles along the x and y dimensions, respectively, using a certain step size (resolution). Rectangles of size 1 (pixels) are not split further. The root rectangle is equipped with a single sum node, representing the distribution over all variables. Each non-root rectangle $\mathcal{R}$, containing more than one variable, is equipped with ρ sum nodes, representing ρ mixture distributions over the variables contained in $\mathcal{R}$. Each rectangle containing exactly one variable is equipped with γ *Gaussian* probability density nodes, which are the leaves of the SPN. The means of the Gaussian nodes are set to the γ quantile means of the corresponding variables, calculated from the training set, and the standard deviation is set to 1. If $\mathcal{R}'$ and $\mathcal{R}''$ are two rectangles generated by some split of $\mathcal{R}$, then for each combination of nodes N', N'', where N' comes from $\mathcal{R}'$ and N'' comes from $\mathcal{R}''$, a product node is generated and connected as parent of N' and N''. The so-generated product nodes are connected as child of each sum node in $\mathcal{R}$. The weights of this SPN are trained by a type of hard (winner-take-all) EM, with a sparseness penalty, penalizing evocation of non-zero weights.

In [10], SPNs were trained for image recognition using *conditional* likelihood, i.e. a discriminative criterion. In [11, 8, 7], algorithms were proposed which do not rely on rectangular organization of data. Closely related to SPNs are arithmetic circuits (ACs). In [5, 12], ACs were learned to represent graphical models with tractable inference. In [9], the algorithm proposed in [8] was modified to learn SPNs over distributions represented by ACs.

3. BANDWIDTH EXTENSION USING SUM-PRODUCT NETWORKS

In [6], SPNs were used to recover missing (covered) parts of face images. Translated to the audio domain, specifically to the ABE problem, this corresponds to recover high frequencies from the telephone band. In this paper, we modify the HMM-based framework for ABE [13, 14] and incorporate SPNs for modeling the observations.

In the HMM-based system [13] time signals are processed in frames with some overlap, yielding a total number of T frames. For each frame, the spectral envelope of the high-band is modeled using cepstral coefficients obtained from linear prediction (LP). On a training set, these coefficients are clustered using the LBG algorithm [15]. The temporally ordered cluster indices are used as hidden state sequence of an HMM, whose prior and transition probabilities can be estimated using the observed frequency estimates. For each hidden state, an observation GMM is trained on features taken from the low-band (see [13] for details about these features). In the test phase, the high frequency components and therefore the hidden states of the HMM are missing. For each time frame, the marginal probability of the hidden state is inferred using the forward-backward algorithm [3]. For real-time capable systems, the backward-messages have to be obtained from a limited number of $\lambda \geq 0$ look-ahead frames. Using the hidden state posterior, an MMSE estimate of the high-band cepstral coefficients is obtained [13], which together with the periodogram of the low-band yield estimates of the wide-band cepstral coefficients. To extend the excitation signal to the high-band, the low-band excitation is modulated either with a fixed frequency carrier, or with a pitch-dependent carrier. According to [13] and related ABE literature, the results are quite insensitive to the method of extending the excitation.

In this paper, we use the log-spectra of the time frames as observations, where the symmetric, redundant frequency bins are discarded. Let $S(t, f)$ be the f^{th} frequency bin of the t^{th} time-frame of the full-band signal, $t \in \{1, \ldots, T\}$, $f \in \{1, \ldots, F\}$, where F is the number of frequency bins and $\mathbf{S}_t = (S(t, 1), \ldots, S(t, F))^T$. We cluster the log-spectra $\{\mathbf{S}_{1:T}\}$ of training speech using the LBG algorithm, and use the cluster indices as hidden states of an HMM. On each cluster, we train an SPN, yielding state-dependent models over the log-spectra. For training SPNs, we use the algorithm proposed in [6] requiring that the data is organized as rectangular array; here the data is a $1 \times F$ rectangular array. We used $\rho = 20$ sum nodes per rectangle and $\gamma = 20$ Gaussian PDF nodes per variable (see section 2). This values were chosen as an "educated guess" and *not cross-validated*. Similar as in [6], we use a *coarse resolution* of 4, i.e. rectangles of height larger than 4 are split with a stepsize of 4.

For ABE we simulate narrow-band telephone speech [16] by applying a bandpass filter with stop frequencies 50 Hz and 4000 Hz. Let $\bar{S}(t, f)$ be the time-frequency bins of the telephone filtered signal, and $\bar{\mathbf{S}}_t = (\bar{S}(t, 1), \ldots, \bar{S}(t, F))^T$. Within the telephone band, we can assume that $S(t, f) \approx \bar{S}(t, f)$, while some of the lowest and the upper half of the frequency bins in $\bar{\mathbf{S}}_t$ are lost. To perform inference in the HMM, this requires that the missing data is marginalized in the state-dependent models, which can be done efficiently in SPNs [6]. More precisely, Gaussian PDF nodes corresponding to unobserved frequency bins, constantly return value 1. In this way, these variables are marginalized by the SPN in the upward-pass. The output probabilities serve as observation likelihoods and are processed by the forward-backward algorithm [3]. This delivers the marginals $p(Y_t|\mathbf{e}_t)$, where Y_t is the hidden HMM variable in the the t^{th} time frame, and $\mathbf{e}_t$ is the observed data up to time frame t, i.e. all frequency bins in the telephone band, for all time

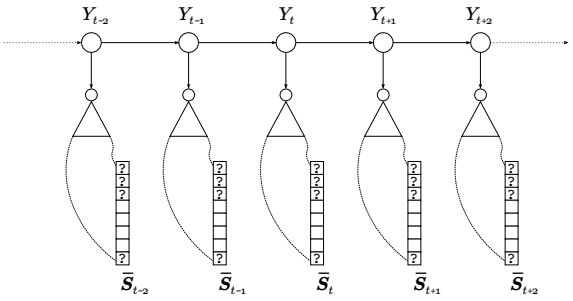


Fig. 1. Illustration of the HMM with SPN observation models. State-dependent SPNs are symbolized by triangles with a circle on top. For the forward-backward algorithm, frequency bins marked with "?" (missing) are marginalized out by the SPNs.

frames $1, \ldots, (t+\lambda)$. An illustration of the modified HMM used in this paper is given in Figure 1. Following [6], we use *most-probable-explanation* (MPE) inference for recovering the missing spectrogram content, where we reconstruct the high-band only. Let $\hat{\mathbf{S}}_{t,k} = (\hat{S}_{t,k}(1), \ldots \hat{S}_{t,k}(F))^T$ be the MPE-reconstruction of the t^{th} time frame, using the SPN depending on the k^{th} HMM-state. Then we use the following bandwidth-extended log-spectrogram

$$\hat{S}(t,f) = \begin{cases} \bar{S}(t,f) & \text{if } f < f' \\ \sum_{k=1}^{K} p(Y_t = k|\mathbf{e}_t)\hat{S}_{t,k}(f) & \text{o.w.} \end{cases} \quad (1)$$

where f' corresponds to $4000\,\text{Hz}$.

4. RECONSTRUCTING TIME SIGNALS

To synthesize a time-signal from the bandwidth extended log-spectrogram, we need to associate a phase to the estimated magnitude spectrogram $e^{\hat{S}(t,f)}$. The problem of recovering a time-domain signal given a modified magnitude appears in many speech applications, such as single-channel speech enhancement [17, 18, 19], single-channel source separation [20, 21, 22, 23] and speech signal modification [24, 25]. These signal modifications are solely employed in spectral amplitude domain while the phase information of the desired signal is not available. A typical approach is to use the observed (noisy) phase spectrum or to replace it with an enhanced/estimated phase.

In order to recover phase information for ABE, we use the iterative algorithm proposed by Griffin and Lim (GL) [26]. Let $j \in \{0, \ldots, J\}$ be an iteration index, and $\hat{C}^{(j)}$ be a complex valued matrix generated in the j^{th} iteration. For $j = 0$, we have

$$\hat{C}^{(0)}(t,f) = \begin{cases} \bar{C}(t,f) & 1 \le f \le f' \\ e^{\hat{S}(t,f)} & \text{o.w.} \end{cases} \quad (2)$$

where $\bar{C}$ is the complex spectrogram of the bandpass filtered input signal. Within the telephone band, phase information is considered reliable and copied from the input. Outside of the narrow-band, phase is initialized with zero. Note that in general $\hat{C}^{(0)}$ is *not a valid* spectrogram since a time signal whose STFT equals $\hat{C}^{(0)}$ might not exist. The j^{th} iteration of the GL algorithm is given by

$$\hat{C}^{(j)}(t,f) = \begin{cases} \bar{C}(t,f) & 1 \le f \le f' \\ e^{\hat{S}(t,f)}\, e^{i\angle \mathcal{G}(\hat{C}^{(j-1)})(t,f)} & \text{o.w.} \end{cases} \quad (3)$$

$$\mathcal{G}(C) = \text{STFT}(\text{STFT}^{-1}(C)). \quad (4)$$

At each iteration, the magnitude of the approximate STFT $\hat{C}^{(j)}$ equals the magnitude $e^{\hat{S}}$ estimated by our model, while temporal coherence of the signal is enforced by the operator $\mathcal{G}(\cdot)$ (see e.g. [25] for more details). The estimated time signal s_j at the j^{th} iteration is given by $s_j = \text{STFT}^{-1}\left(\hat{C}^{(j)}\right)$. At each iteration, the mean square error between $|\text{STFT}(s_j)|$ and $|\hat{C}^{(0)}|$ is reduced [26]. In our experiments, we set the number of iterations $J = 100$, which appeared to be sufficient for convergence.

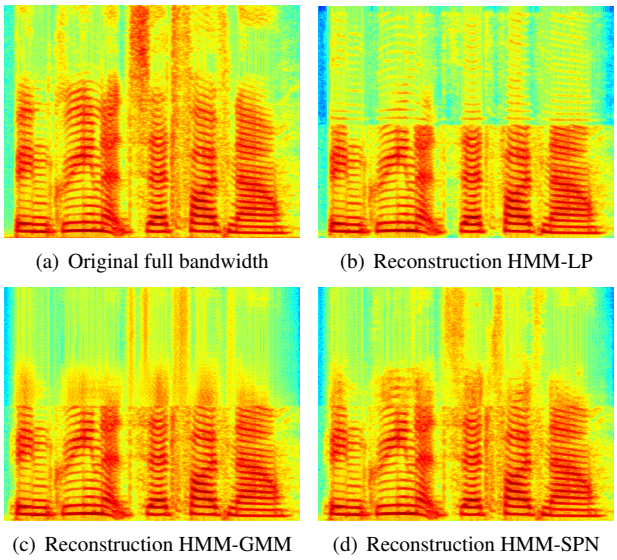
5. EXPERIMENTS

We used 2 baselines in our experiments. The first baseline is the method proposed in [13], based on the vocal tract filter model using linear prediction. We used 64 HMM states and 16 components per state-dependent GMM, which performed best in [13]. We refer as HMM-LP to this baseline. The second baseline is almost identical to our method, where we replaced the SPN with a Gaussian mixture model with 256 components with diagonal covariance matrices. For training GMMs, we ran the EM algorithm for maximal 100 iterations and using 3 random restarts. Inference using the GMM model works the same way as described in section 3, since a GMM can be formulated as an SPN with a single sum node [7]. We refer as HMM-GMM to this baseline. To our method, we refer as HMM-SPN. For HMM-GMM and HMM-SPN, we used the same clustering of log-spectra using a codebook size of 64.

We used time-frames of 512 samples length, with 75% overlap, which using a sampling frequency of $16\,\text{kHz}$ corresponds to a frame length of $32\,\text{ms}$ and a frame rate of $8\,\text{ms}$. Before applying the FFT, the frames were weighted with a Hamming window. For the forward-backward algorithm we used a look-ahead of $\lambda = 3$ frames, which corresponds to the minimal delay introduced by the 75% frame-overlap. We performed our experiments on the GRID corpus [27], where we used the test speakers with numbers 1, 2, 18, and 20, referred to as s1, s2, s18, and s20, respectively. Speakers s1 and s2 are male, and s18 and s20 are female. We trained *speaker dependent* and *speaker independent* models. For speaker dependent models we used 10 minutes of speech of the respective speaker. For speaker independent models we used 10 minutes of speech obtained from the remaining 30 speakers of the corpus, each speaker providing approximately 20 seconds of speech. For testing we used 50 utterances per test speaker, not included in the training set.

Fig. 2 shows log-spectrograms of a test utterance of speaker s18 and the bandwidth extended signals by HMM-LP, HMM-GMM and HMM-SPN, using speaker dependent models. We see that HMM-LP succeeds in reconstructing a harmonic structure for voiced sounds. However, we see that fricative and plosive sounds are not well captured. The reconstruction by HMM-GMM is blurry and does not recover the harmonic structure of the original signal well, but partly recovers high-frequency content related to consonants. The HMM-SPN method recovers a natural high frequency structure, which largely resembles the original full-band signal: the harmonic structure appears more natural than the one delivered by HMM-LP and consonant sounds seem to be better detected and reconstructed than by HMM-GMM. According to informal listening tests[1], the visual impression corresponds to the listening experience: the signals delivered by HMM-SPN clearly enhance the high-frequency content and sound more natural than the signals delivered by HMM-LP and

[1] Formal listening tests were out of the scope of the paper. All ABE signals, the full-band and the narrow-band telephone signals can be obtained as WAV files from http://www2.spsc.tugraz.at/people/peharz/ABE/



(a) Original full bandwidth (b) Reconstruction HMM-LP

(c) Reconstruction HMM-GMM (d) Reconstruction HMM-SPN

Fig. 2. Log-spectrogram of the utterance "Bin green at zed 5 now", spoken by s18. (a): original full bandwidth signal. (b): ABE result of HMM-LP [13]. (c): ABE result of HMM-GMM (this paper). (d): ABE results of HMM-SPN (this paper).

Table 1. Average LSD using speaker-dependent models.

	s1	s2	s18	s20
HMM-LP	7.13	7.57	6.48	6.41
HMM-GMM	3.18	2.93	2.28	2.82
HMM-SPN	**3.12**	**2.84**	**2.15**	**2.59**

HMM-GMM. HMM-GMM and HMM-SPN both deliver a more realistic extension for fricative and plosive sounds. However, this introduces also a some high frequency noise. According to our listening experience, these artifacts are less severe for the HMM-SPN signals.

For an objective evaluation, we use the log-spectral distortion (LSD) in the high-band [13]. Given an original signal and an ABE reconstruction, we perform L^{th}-order LPC analysis for each frame, where $L = 9$. This yields $(L+1)$-dimensional coefficient vectors $\mathbf{a}_\tau$ and $\hat{\mathbf{a}}_\tau$ of the original and the reconstructed signals, respectively, where τ is the frame index. The spectral envelope modeled by a generic LPC coefficient vector $\mathbf{a} = (a_0, \ldots, a_L)^t$ is given as

$$E_{\mathbf{a}}(e^{j\Omega}) = \frac{\sigma}{|\sum_{k=0}^{L} a_k e^{-jk\Omega}|}, \tag{5}$$

where σ is the square-root of the variance of the LPC-analyzed signal. The LSD for the τ^{th} frame, in high-band is calculated as

$$\text{LSD}_\tau = \sqrt{\frac{\int_\nu^\pi \left(20 \log E_{\mathbf{a}_\tau}(e^{j\Omega}) - 20 \log E_{\hat{\mathbf{a}}_\tau}(e^{j\Omega})\right)^2 d\Omega}{\pi - \nu}}, \tag{6}$$

where $\nu = \pi \frac{4000}{\text{fs}/2}$, fs being the sampling frequency. The LSD at utterance level is given as the average of LSD_τ over all frames.

Tables 1 and 2 show the LSD of all three methods for the speaker dependent and speaker independent scenarios, respectively, averaged over the 50 test sentences. For each speaker, we see a clear ranking of the three method, and that the HMM-SPN method always performs best. All differences are significant at a 0.95 confidence level, according to a paired one-sided t-test.

Table 2. Average LSD using speaker-independent models.

	s1	s2	s18	s20
HMM-LP	7.12	7.66	6.60	6.34
HMM-GMM	3.62	4.46	3.82	3.60
HMM-SPN	**3.42**	**3.85**	**3.05**	**3.36**

6. DISCUSSION

We demonstrated that SPNs are a promising probabilistic model for speech, applying them to the ill-posed problem of artificial bandwidth extension. Motivated by the success of SPNs on the also ill-posed and related problem of image completion, we used SPNs as observation models in HMMs, modeling the temporal evolution of log short-time spectra. While the model is trained on full-band speech, the fact that the high and very low frequencies are missing in telephone signals is naturally treated by marginalization of missing frequency bins. Recovering the missing high frequencies, is naturally treated by MPE inference. The resulting system clearly improves the state of the art both in subjective listening tests and objective performance evaluation using the log-spectral distortion measure.

This performance improvement comes at an increased computational cost. The trained observation SPNs have 136 layers and tens of thousand of nodes and parameters. Therefore, bandwidth extension using our HMM-SPN approach currently takes about $1-2$ minutes computation time per utterance on a standard desktop computer, using a non-optimized Matlab/C++-based prototype. Inference using the HMM-GMM model requires approximately $0.5-1$ minutes per utterance; inference in the HMM-LP model requires some seconds. Therefore, although we designed the overall system to be real-time capable (small HMM look-ahead), it is currently not suitable for a real-time application implemented on a low-energy embedded system. For non-real-time systems, e.g. for offline processing of telephone speech databases, the approach presented here is appropriate. The basic motivation in this paper, however, was to demonstrate the applicability of SPNs for modeling speech; according to prior studies [6, 8], SPNs are able to express complex interaction with comparable little inference time. Therefore one can conjecture that an ABE system with classical graphical models, expressing a similar amount of dependencies as the used SPNs, would have an overall computation time in the range of hours.

The system presented in this paper is trained in a two-step approach, i.e. (i) clustering the training data which delivers the HMM states and statistics, and (ii) subsequent training of state-dependent observation models. Incorporating state-sequence modeling directly into SPN training, similar as in dynamic graphical models, is a interesting future research direction. Finally, future directions for research on SPN-based speech models are further speech related applications, such as packet loss concealment, (single channel) source separation, and speech enhancement.

7. REFERENCES

[1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.

[2] F. Pernkopf, R. Peharz, and S. Tschiatschek, *Introduction to Probabilistic Graphical Models*, vol. 1 of *Academic Press Library in Signal Processing*, chapter 18, Elsevier, 2013.

[3] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, vol. 77, pp. 257–286.

[4] A. Darwiche, "A Differential Approach to Inference in Bayesian Networks," *ACM*, vol. 50, no. 3, pp. 280–305, 2003.

[5] D. Lowd and P. Domingos, "Learning arithmetic circuits," in *Uncertainty in Artificial Intelligence*, 2008, pp. 383–392.

[6] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *Uncertainty in Artificial Intelligence*, 2011, pp. 337–346.

[7] R. Peharz, B. Geiger, and F. Pernkopf, "Greedy Part-Wise Learning of Sum-Product Networks," in *ECML/PKDD*. 2013, vol. 8189, pp. 612–627, Springer Berlin.

[8] R. Gens and P. Domingos, "Learning the Structure of Sum-Product Networks," in *ICML*, 2013, pp. 873–880.

[9] A. Rooshenas and D. Lowd, "Learning sum-product networks with direct and indirect variable interactions," *ICML – JMLR W&CP*, vol. 32, pp. 710–718, 2014.

[10] R. Gens and P. Domingos, "Discriminative learning of sum-product networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 3248–3256.

[11] A. Dennis and D. Ventura, "Learning the architecture of sum-product networks using clustering on variables," in *NIPS*, 2012, pp. 2042–2050.

[12] D. Lowd and A. Rooshenas, "Learning markov networks with arithmetic circuits," *Proceedings of AISTATS*, pp. 406–414, 2013.

[13] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, pp. 1707–1719, 2003.

[14] G.-B. Song and P. Martynovich, "A study of HMM-based bandwidth extension of speech signals," *Signal Processing*, vol. 89, pp. 2036–2044, 2009.

[15] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Transaction on Communication*, vol. 28, no. 1, pp. 84–95, 1980.

[16] "ETSI: Digital cellular telecommunications system (phase 2+); enhanced full rate (EFR) speech transcoding, ETSI EN 300 726 v8.0.1," Nov. 2000.

[17] C. Leitner and F. Pernkopf, "Speech enhancement using pre-image iterations," in *ICASSP*, 2012, pp. 4665–4668.

[18] P. Mowlaee and R. Saeidi, "Iterative closed-loop phase-aware single-channel speech enhancement," *Signal Processing Letters, IEEE*, vol. 20, no. 12, pp. 1235–1239, 2013.

[19] P. Mowlaee and R. Saeidi, "On phase importance in parameter estimation in single-channel speech enhancement," in *ICASSP*, 2013, pp. 7462–7466.

[20] R. Peharz, M. Stark, and F. Pernkopf, "A factorial sparse coder model for single channel source separation," in *Interspeech*, 2010, pp. 386–389.

[21] M. Stark, M. Wohlmayr, and F. Pernkopf, "Source-filter based single channel speech separation using pitch information," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 2, pp. 242–255, 2011.

[22] P. Mowlaee, R. Saiedi, and R. Martin, "Phase estimation for signal reconstruction in single-channel speech separation," in *ICSLP*, 2012.

[23] M. K. Watanabe and P. Mowlaee, "Iterative sinusoidal-based partial phase reconstruction in single-channel source separation," in *ICSLP*, 2013, pp. 832–836.

[24] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: a state of the art," in *DAFX*, 2011, pp. 375–386.

[25] J. Le Roux, *Exploiting Regularities in Natural Acoustical Scenes for Monaural Audio Signal Estimation, Decomposition, Restoration and Modification*, Ph.D. thesis, The University of Tokyo & Université Paris, 2009.

[26] D. Griffin and J.S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[27] M.P. Cooke, J. Barker, S.P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Amer.*, vol. 120, pp. 2421–2424, Nov 2005.