Abstract—The performance of a speaker recognition system degrades considerably in the presence of noise. One approach to significantly increase robustness is to use the marginal probability density function of the spectral features that reliably represent speech. Current state-of-the-art speaker recognition systems employ non-probabilistic models, such as convolutional neural networks (CNNs), which cannot use marginalisation. As an alternative, we propose the use of sum-product networks (SPNs), a deep probabilistic graphical model which is compatible with marginalisation. SPN speaker models are evaluated here on real-world non-stationary and coloured noise sources at multiple SNR levels. In terms of speaker recognition accuracy, SPN speaker models employing marginalisation are more robust than recent CNN-based speaker recognition systems that pre-process the noisy speech. Additionally, the SPN speaker models consist of significantly fewer parameters than that of the CNN-based speaker recognition systems. The results presented in this work show that SPN speaker models are a robust, parameter-efficient alternative for speaker recognition.

Availability: The SPN speaker recognition system is available at: https://github.com/anicolson/SPN-Spk-Rec

Index Terms—Sum-product network (SPN), marginalisation, missing feature approach, robust speaker recognition.

I. INTRODUCTION

The task of a text-independent speaker recognition system is to identify a speaker from a given voice recording, irrespective of its linguistic content. This is accomplished by modelling the voice characteristics of each speaker after an enrolment phase [1]. While the most common application of speaker recognition is verification [2], other important applications exist [3]. One role that a speaker recognition system can fill is the selection of a speaker-dependent acoustic model for a speech recognition system [4]. It can also be used to perform speaker segmentation, an important pre-processing step for speaker diarisation [5]. The realisation of each application is dependent upon a high-performance speaker recognition system. The first speaker recognition system to have been considered as high performance modelled each speaker’s voice using a Gaussian mixture model (GMM) [6].

One obstacle that prevented the commercial introduction of GMM speaker models was their poor performance in the presence of noise [7], spurring the investigation of robust approaches [8]. A noteworthy approach was the missing feature approach, which is underpinned by evidence that speech is intelligible to humans even after it has undergone substantial spectral masking [9]. Marginalisation, as proposed by Cook et al. [10], has been the most prominent missing feature approach in the literature [11], and is able to significantly increases the robustness of a GMM speaker model [12]. With marginalisation, the marginal probability density function is obtained by integrating over the components of the feature vector that have been classified as unreliable representations of speech [13]. Classification is thus performed on a partial instantiation of a given feature vector, consisting of only the components that reliably represent speech.

Recently, speaker recognition has been performed using convolutional neural networks (CNNs) [14], which provide a significant improvement over GMM speaker models for clean conditions. One recent example is SincNet, which employs parametrised sinc functions to pre-define a bank of band-pass filters for its first layer [15]. Another example, as proposed by Xie et al. [16], uses a ‘thin’ residual CNN, and dictionary-based NetVLAD [17] and GhostVLAD [18] layers for feature aggregation (referred to as Xie2019 henceforth). Despite their high performance on clean speech, modern speaker recognition systems are still susceptible to performance degradation in the presence of noise [19]. CNNs are however not probabilistic models and are incapable of inferring from partial instantiations of a given feature vector, meaning that they cannot employ classifier-compensation missing feature approaches like marginalisation. Currently, the most popular approach to increase their robustness is to use a front-end technique to pre-process the noisy speech [20].

In 2011, Poon et al. [22] proposed a model that is both a deep architecture and a tractable probabilistic graphical model, called a sum-product network (SPN). For the deep architecture case, an SPN can be described as a deep neural network, restricted to using sum and product operators. For the probabilistic graphical model case, an SPN can be described as a rooted directed acyclic graph, with variables as leaves. SPNs have clear semantics; each node represents an unnormalised joint probability distribution over a set of variables. As an SPN is a probabilistic model, it can perform inference on a partial instantiation of a given feature vector, making it applicable for marginalisation.

Here, SPNs utilising marginalisation are proposed as robust speaker models. SPN speaker models have both advantages and disadvantages over CNN speaker recognition systems. SPN speaker models are trained solely on clean speech, and can use marginalisation to remain robust, avoiding the need for the noisy speech to be pre-processed. When a new speaker is enrolled, their SPN speaker model is simply added to the set of already existing SPN speaker models, whereas a CNN must be retrained for all speakers. Additionally, the accuracy
of classifying the reliability of a spectral feature has increased in recent years [13], further supporting the case for robust SPN speaker models utilising marginalisation. A disadvantage is that SPN structure and weight learning algorithms, as well as libraries, are currently undeveloped, as highlighted by Jaini et al. [23].

SPN speaker models are evaluated here on a text-independent speaker recognition task against both GMM speaker models [24], SincNet [15], and Xie2019 [16]. Both the SPN and GMM speaker models employ marginalisation, whilst SincNet and Xie2019 employ the long short-term memory ideal ratio mask (LSTM-IRM) estimator by Chen et al. [23]. For bounded memory ideal ratio mask (LSTM-IRM) estimator by Chen et al. [25] as its front-end. Speaker recognition accuracy is used in recent years [13], further supporting the case for robust

A. Frequency Domain Representation

A frequency domain representation is used as the feature vector for marginalisation, like the power spectral density (PSD) estimate of the short-time Fourier transform (STFT), or the spectral sub-band energies of the PSD estimate. In this work, we use the log-spectral sub-band energies (LSSEs) of a clean speech frame as the feature vector for the SPN and GMM speaker models. The LSSEs are computed from the single-sided PSD estimate of the STFT of the clean speech using the periodogram method, as in [12].

\[
X_b = \log \sum_{k=0}^{N_t/2} h_{b,k} \hat{P}_k, \quad 0 \leq b \leq B - 1, \tag{1}
\]

where \(N_t\) denotes the frame length in discrete-time samples, \(k\) denotes the discrete-frequency index, \(\hat{P}_k\), for all \(k\), denotes the PSD estimate for a frame, and \(h_{b,k}\), for all \(k\), denotes the \(b\)th filter of a bank of \(B\) triangular-shaped critical band filters spaced uniformly on the mel-scale. The PSD is estimated from the STFT of the clean speech using the periodogram method, as in [12].

B. SPN Speaker Models with Gaussian Leaves

An SPN [22] specifies an unnormalised joint distribution over a set of random variables, \(X = (X_1, X_2, ..., X_B)^T\), where in this case, \(X\) is the feature vector of LSSEs for a frame of clean speech. An observation of \(X\) is denoted here by \(x = (x_1, x_2, ..., x_B)^T\). Hence, the SPN, \(S\), for speaker class \(C\) is a function of the observed feature vector: \(S(x|C)\), where the value of the SPN is given by its root. An SPN consists of multiple layers of sum and product nodes, with distributions as leaves. The multivariate distribution of the \(i\)th leaf is over a subset of the variables: \(X_i \subseteq X\), and is assumed to be normally

\[
S(x|C) = \prod_{d \in D} \frac{1}{\sqrt{2\pi \Sigma_{d,d}^{(2)}}} e^{-\frac{(x_d - \mu_d)^2}{2\Sigma_{d,d}^{(2)}}}, \tag{2}
\]

where \(D \subseteq \{1, 2, ..., B\}^T\), and indicates the random variable indices for \(X_i\). An SPN over two variables with univariate Gaussian leaves is shown in Figure 1.

If node \(i\) is a product node, its value is given by the product of the values of its children, \(Ch(.)\): \(S_i = \prod_{j \in Ch(S_i)} S_j\), where \(S_j\) is the \(j\)th child of node \(S_i\). If node \(i\) is a sum node, its value is given by the sum of the values of its children: \(S_i = \sum_{j \in Ch(S_i)} w_{ij} S_j\), where weight \(w_{ij}\) is the non-negative weighted edge between \(S_i\) and \(S_j\), where \(j \in Ch(S_i)\). To be a valid joint distribution, an SPN must be both decomposable, and complete, as described in [22]. The scope of a node, \(Sc(.)\), is defined as the set of variables that are descendants of it. An SPN is said to be decomposable when the scopes of the children of its product nodes are disjoint: \(\forall S_j, S_k \in Ch(S_i), Sc(S_j) \cap Sc(S_k) = \emptyset\), where \(\emptyset\) indicates an empty set. An SPN is said to be complete when the scopes of the children of its sum nodes are identical: \(\forall S_j, S_k \in Ch(S_i), Sc(S_j) = Sc(S_k)\).

C. Marginalisation for SPNs

For marginalisation, each component of an observed noisy speech feature vector is classified as either a reliable or an unreliable representation of the corresponding unobserved clean speech feature vector components. The noisy speech feature vector, \(y\), can thus be described as the union of the reliable and unreliable components: \(y = y^r \cup y^u\). Here, we not only apply marginalisation to SPNs, but also bounded marginalisation, as proposed by Cook et al. [22]. For bounded marginalisation, the unreliable components are treated as the upper bounds to the unobserved clean speech component values. For LSSEs, the bounds are taken from \([-\infty, y^u]\). Thus, the probability density function for the \(i\)th leaf becomes

\[
N(y^r_i, x^u_i \leq y^u_i | i, C) = \int_{-\infty}^{y^r_i} N(x^u_i | i, C) dx^u_i. \tag{3}
\]
TABLE I

Speaker recognition accuracy for the real-world non-stationary noise sources. The average improvement over the model in the preceding row is shown in the last column. The highest accuracy for each condition is shown in boldface.

| Model       | Marg. | Bounds | SNR level (dB) | Average impr. |
|-------------|-------|--------|----------------|---------------|
|             |       |        | Voice babble   | Street music  |               |
|             |       |        | -5 0 5 10 15   | -5 0 5 10 15  |               |
| GMM[24]     | no    | no     | 0.00 0.00 0.63 13.02 50.48 | 0.00 0.00 0.95 5.40 25.40 | -           |
| SPN         | no    | no     | 0.00 0.00 1.59 15.56 50.16 | 0.00 0.32 1.27 6.03 25.71 | +0.48       |
| GMM[24]     | yes   | no     | 2.22 6.35 18.10 46.98 79.37 | 4.76 10.48 20.32 37.46 66.35 | +2.16       |
| SPN         | yes   | no     | 2.22 7.30 19.05 50.79 83.49 | 4.13 10.48 24.13 40.95 71.43 | +2.16       |
| GMM[24]     | yes   | yes    | 15.24 29.21 48.57 72.70 89.21 | 20.63 32.06 54.60 71.11 85.40 | -           |
| SPN         | yes   | yes    | 14.60 32.70 55.87 77.78 91.43 | 22.54 34.92 59.37 74.29 90.16 | +3.49       |
| SincNet[15] | -     | -      | 0.63 1.59 18.10 56.83 93.02 | 0.63 2.86 11.11 46.98 85.40 | -6.54       |
| SincNet[15] + IRM[25] | -     | -      | 0.63 1.59 18.10 56.83 93.02 | 0.63 2.86 11.11 46.98 85.40 | -6.54       |
| Xie2019[16] + IRM[25] | -     | -      | 0.63 1.59 18.10 56.83 93.02 | 0.63 2.86 11.11 46.98 85.40 | -6.54       |
| Xie2019[16] | -     | -      | 0.63 1.59 18.10 56.83 93.02 | 0.63 2.86 11.11 46.98 85.40 | -6.54       |

For marginalisation, the unreliable components are treated as missing. Thus, the bounds are taken from \([-\infty, \infty]\), and the integral in Equation [3] reduces to unity. This gives the marginal probability density function for the leaf: \(N(y_i^r | i, C)\). When all of the components of \(y_i\) are unreliable, it is treated as a vector with no instantiated components: \(N(y_i^r = \emptyset | i, C) = 1\).

III. EXPERIMENT SETUP

A. Signal Processing

The feature vectors for the GMM and SPN speaker models were computed using the following hyperparameters. The Hamming window function was used for analysis, with a frame length of 32 ms (512 discrete-time samples) and a frame shift of 16 ms (256 discrete-time samples). The 257-point single-sided PSD estimate for a frame included both the DC and Nyquist frequency component. The LSSEs of a PSD estimate were computed using the following hyperparameters. The Hamming window function was used for analysis, with a frame length of 32 ms (512 discrete-time samples) and a frame shift of 16 ms (256 discrete-time samples). The 257-point single-sided PSD estimate for a frame included both the DC and Nyquist frequency component. The LSSEs were computed using 26 triangular-shaped critical band filters spaced uniformly on the mel-scale.

B. Classification of Reliable Spectral Components

Here, the reliability of a spectral component is determined by its a priori SNR, as in [26]. A component with an a priori SNR of greater than 0 dB is classified as reliable [27]. Deep Xi from [28] is used here as the a priori SNR estimator. It is a deep learning approach to a priori SNR estimation, and is available at: https://github.com/anicolson/DeepXi. Deep Xi estimates the a priori SNR for each of the 257 frequency-domain components of a noisy speech frame. The a priori SNR estimate for each sub-band is subsequently found by applying the filterbank used to compute the LSSEs.

C. Training and Testing Sets

The TIMIT corpus [29] (16 kHz, single-channel), which consists of 630 speakers with 10 utterances each, was used as the clean speech set in this work. The si* and sz* subsets were used for training (5,040 utterances) and the so* subset was used for testing (1,260 utterances). Each clean speech recording from the so* subset was mixed additionally with one of four real-world noise source recordings to create the noisy speech for testing (315 clean speech recordings for each noise source). Each noisy speech recording was replicated at five SNR levels: -5 to 15 dB, in 5 dB increments, forming a testing set of 6,300 noisy speech recordings. The real-world noise sources included two non-stationary and two coloured. The two real-world non-stationary noise sources included voice babble from the RSG-10 noise dataset [30] and street music from the Urban Sound dataset [31]. The two real-world coloured noise sources included F16 and factory (welding) from the RSG-10 noise dataset [30].

D. Speaker Model Configurations

GMM: For each speaker, a GMM consisting of 48 diagonal covariance clusters was trained on the training set using the expectation-maximisation algorithm [32], and the k-means++ algorithm for parameter initialisation [33].

SincNet: [15] is available at: https://github.com/mravanelli/SincNet and was trained using the training set.

Xie2019: [16] is available at: https://github.com/WeidiXie/VGG-Speaker-Recognition and was trained using the training set with an input spectrogram size of 1 second.

SincNet + IRM & Xie2019 + IRM: The LSTM-IRM estimator from [25] was used as the front-end for SincNet and Xie2019. The training data and configuration from [34] was used specifically.

SPN: Each speaker was modelled using an SPN with univariate Gaussian leaves. The SPFlow library was used to implement the SPN speaker models [35]. A variant of the LearnSPN algorithm [36] that partitions and clusters variables was used specifically.

C. Training and Testing Sets

The TIMIT corpus [29] (16 kHz, single-channel), which consists of 630 speakers with 10 utterances each, was used as the clean speech set in this work. The si* and sz* subsets were used for training (5,040 utterances) and the so* subset was used for testing (1,260 utterances). Each clean speech recording from the so* subset was mixed additionally with one of four real-world noise source recordings to create the noisy speech for testing (315 clean speech recordings for each noise source). Each noisy speech recording was replicated at five SNR levels: -5 to 15 dB, in 5 dB increments, forming a testing set of 6,300 noisy speech recordings. The real-world noise sources included two non-stationary and two coloured. The two real-world non-stationary noise sources included voice babble from the RSG-10 noise dataset [30] and street music from the Urban Sound dataset [31]. The two real-world coloured noise sources included F16 and factory (welding) from the RSG-10 noise dataset [30].

IV. RESULTS AND DISCUSSION

A. Real-World Non-Stationary Noise Sources

Table I shows the speaker recognition accuracy for the real-world non-stationary noise sources: voice Babble and
Table II

Speaker recognition accuracy for the real-world coloured noise sources. The average improvement over the model in the preceding row is shown in the last column. The highest accuracy for each condition is shown in boldface.

| Model          | Marg. | Bounds | SNR level (dB) | F16 Factory | Average impr. |
|----------------|-------|--------|----------------|-------------|---------------|
|                |       |        | 5  10  15      | 5  10 15    |               |
| GMM[24]        | no    | no     | 0.32 0.32 0.32 | 0.32 0.32   | 0.32 0.32     |
| SPN            | yes   | no     | 1.90 7.30 21.27 | 3.17 5.71   | 10.16         |
| GMM[24]        | yes   | no     | 1.90 10.16 21.59 | 3.17 5.71   | 13.65         |
| SincNet[15]    | yes   | yes    | 0.32 0.32 0.32 | 0.32 0.32   | 0.32 0.32     |
| Xie2019[16] + IRM[25] | yes | yes | 0.32 0.32 0.32 | 0.32 0.32   | 0.32 0.32     |
| Xie2019[16]    | yes   | no     | 0.32 0.32 0.32 | 0.32 0.32   | 0.32 0.32     |

Table III

Average number of parameters used by each speaker recognition system for each of the 630 speakers.

|            | SPN  | GMM  | Xie2019 | SincNet |
|------------|------|------|---------|---------|
| Params. per speaker | 2502 | 2544 | 13545   | 36718   |

B. Real-World Coloured Noise Sources

Table II shows the speaker recognition accuracy for the real-world coloured noise sources: F16 and factory. The SPN speaker models utilising bounded marginalisation again outperformed SincNet + IRM, with an average performance increase of 21.52%. Over all of the tested conditions in Table II the SPN speaker models demonstrated an average improvement of 1.33% and 2.66% over the GMM speaker models when marginalisation and bounded marginalisation were used, respectively. This indicates that marginalisation and bounded marginalisation are more suited to SPN speaker models than GMM speaker models.

The results presented in Tables I and II show that the SPN speaker models are more robust to both real-world non-stationary and coloured noise sources, when either marginalisation or bounded marginalisation is used. These results are more significant when considering the number of parameters that each speaker recognition system expends on a speaker, as specified in Table III. The SPN speaker models were more robust than SincNet, whilst employing 14.7 times fewer parameters on average per speaker. This exhibits the parameter efficiency of the SPN speaker models.

C. Future Direction

The SPN structure learning algorithm used here, LearnSPN (introduced in 2013) [35], was the second-ever proposed. Several structure learning algorithms that can outperform LearnSPN have since been introduced, including Prometheus [23]. Additionally, it is common to use backpropagation to fine-tune the weights of an SPN (as used to train CNNs), something that was not carried out here. With a better structure learning algorithm, and the use of a weight learning algorithm, the joint distribution of a speaker could perhaps be more effectively modelled. This may improve the performance of SPN speaker models at higher SNR levels. SPN acoustic models utilising marginalisation should also be investigated for robust automatic speech recognition (ASR), and for robust speaker verification, where the universal background model framework could be employed [2].

V. Conclusion

Here, sum-product networks (SPNs) are employed as robust speaker models. They are evaluated on real-world non-stationary and coloured noise sources at multiple SNR levels. When marginalisation is used, SPN speaker models are more robust than current speaker recognition systems that employ significantly more parameters and pre-processing techniques. With the development of better structure and weight learning algorithms, SPNs are predicted to have a bright future not only for robust speaker recognition, but also for robust ASR.
