Speech enhancement with variational autoencoders and alpha-stable distributions
Simon Leglaive, Umut Şimşekli, Antoine Liutkus, Laurent Girin, Radu Horaud

To cite this version:
Simon Leglaive, Umut Şimşekli, Antoine Liutkus, Laurent Girin, Radu Horaud. Speech enhancement with variational autoencoders and alpha-stable distributions. ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing, May 2019, Brighton, United Kingdom. pp.541-545, 10.1109/ICASSP.2019.8682546. hal-02005106

HAL Id: hal-02005106
https://inria.hal.science/hal-02005106
Submitted on 8 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
SPEECH ENHANCEMENT WITH VARIATIONAL AUTOENCODERS AND ALPHA-STABLE DISTRIBUTIONS

Simon Leglaive¹, Umut Şimşekli², Antoine Liutkus³, Laurent Girin¹⁴, Radu Horaud¹

¹Inria Grenoble Rhône-Alpes, France, ²LTCI, Télécom ParisTech, Université Paris-Saclay, France
³Inria and LIRMM, France, ⁴Univ. Grenoble Alpes, Grenoble INP, GIPSA-lab, France

ABSTRACT

This paper focuses on single-channel semi-supervised speech enhancement. We learn a speaker-independent deep generative speech model using the framework of variational autoencoders. The noise model remains unsupervised because we do not assume prior knowledge of the noisy recording environment. In this context, our contribution is to propose a noise model based on alpha-stable distributions, instead of the more conventional Gaussian non-negative matrix factorization approach found in previous studies. We develop a Monte Carlo expectation-maximization algorithm for estimating the model parameters at test time. Experimental results show the superiority of the proposed approach both in terms of perceptual quality and intelligibility of the enhanced speech signal.

Index Terms— Speech enhancement, variational autoencoders, alpha-stable distribution, Monte Carlo expectation-maximization.

1. INTRODUCTION

Speech enhancement is one of the central problems in audio signal processing [1]. The goal is to recover a clean speech signal after observing a noisy mixture. In this work, we address single-channel speech enhancement, which can be seen as an under-determined source separation problem, where the sources are of different nature.

One popular statistical approach for source separation combines a local Gaussian model of the time-frequency signal coefficients with a variance model [2]. In this framework, non-negative matrix factorization (NMF) techniques have been used to model the time-frequency-dependent signal variance [3, 4]. Recently, discriminative approaches based on deep neural networks (DNNs) have also been investigated for speech enhancement, with the aim of estimating either clean spectrograms or time-frequency masks, given noisy spectrograms [5, 6, 7]. As a representative example, a DNN is used in [6] to map noisy speech log-power spectrograms into clean speech log-power spectrograms.

Even more recently, generative models based on deep learning, and in particular variational autoencoders (VAEs) [8], have been used for single-channel [9, 10] and multi-channel speech enhancement [11, 12]. These generative model-based approaches provide important advantages and justify the interest of semi-supervised methods for speech enhancement. Indeed, as shown in [9, 10], fully-supervised methods such as [6] may have issues for generalizing to unseen noise types. The method proposed in [10] was shown to outperform both a semi-supervised NMF baseline and the fully-supervised deep learning approach [6].

In most cases, probabilistic models for source separation or speech enhancement rely on a Gaussianity assumption, which turns out to be restrictive for audio signals [13]. As a result, heavy-tailed distributions have started receiving increasing attention in the audio processing community [14, 15, 16]. In particular, α-stable distributions (cf. Section 3) are becoming popular heavy-tailed models for audio modeling due to their nice theoretical properties [17, 13, 14, 18, 19, 20, 21].

In this work, we investigate the combination of a deep learning-based generative speech model with a heavy-tailed α-stable noise model. The rationale for introducing a noise model based on heavy-tailed distributions as opposed to a structured NMF approach as in [10] is to avoid relying on restricting assumptions regarding stationarity or temporal redundancy of the noisy environment, that may be violated in practice, leading to errors in the estimates. In addition, we let the noise model remain unsupervised in order to avoid the aforementioned generalization issues regarding the noisy recording environment. We develop a Monte Carlo expectation-maximization algorithm [22] for performing maximum likelihood estimation at test time. Experiments performed under challenging conditions show that the proposed approach outperforms the competing approaches in terms of both perceptual quality and intelligibility.

2. SPEECH MODEL

We work in the short-term Fourier transform (STFT) domain where \( \mathbb{B} = \{0, ..., F - 1\} \times \{0, ..., N - 1\} \) denotes the set of time-frequency bins. For \((f,n) \in \mathbb{B}, f \) denotes the frequency index and \( n \) the time-frame index. We use \( s_{f,n}, b_{f,n}, x_{f,n} \in \mathbb{C} \) to denote the complex STFT coefficients of the speech, noise, and mixture signals, respectively.

As in [9, 10], independently for all \((f,n) \in \mathbb{B}\), we consider the following generative speech model involving a latent random vector \( h_n \in \mathbb{R}^L \), with \( L \ll F^2 \):

\[
   h_n \sim \mathcal{N}(0, I);
   s_{f,n} | h_n \sim \mathcal{N}(0, \sigma^2_{s,f}(h_n)),
\]

where \( \mathcal{N}(\mathbf{x}; \mu, \Sigma) \) denotes the multivariate Gaussian distribution for a real-valued random vector, \( I \) is the identity matrix of appropriate size, and \( \mathcal{N}(\mathbf{x}; \mu, \sigma^2) \) denotes the univariate complex proper Gaussian distribution. As represented in Fig. 1a, \( \{\sigma^2_{s,f} : \mathbb{R}^L \rightarrow \mathbb{R}_+\}_{f,n} \) is a set of non-linear functions corresponding to a neural network which takes as input \( h_n \in \mathbb{R}^L \). This variance term can be understood as a model for the short-term power spectral density of speech [23]. We denote by \( \theta \), the parameters of this generative neural network.

An important contribution of VAEs [8] is to provide an efficient way of learning the parameters of such a generative model. Let \( \mathbf{s} = (s_{f,n} \in \mathbb{C}^{F^2})_{f,n} \) be a training dataset of clean-speech STFT time frames and \( \mathbf{h} = (h_n \in \mathbb{R}^L)_{n=0}^N \) the set of associated latent random

This work is supported by the ERC Advanced Grant VHIA #340113.
vectors. Taking ideas from variational inference, VAEs estimate the parameters \( \theta \) by maximizing a lower bound of the log-likelihood \( \ln p(s; \theta, \theta_\theta) \) defined by:
\[
\mathcal{L}(\theta_s, \psi) = \mathbb{E}_{q(h|s, \psi)}[\ln p(s | h; \theta_s)] - D_{KL}(q(h | s; \psi) \| p(h)),
\]
where \( q(h | s; \psi) \) denotes an approximation of the intractable true posterior distribution \( p(h | s; \theta_s) \), and \( D_{KL}(q(p) \| p) = \mathbb{E}_q[\ln(q/p)] \) is the Kullback-Leibler divergence.

For generating network, the set of observed and latent variables at a given time frame \( n \) is defined by:
\[
h_{t,n} | s_n \sim \mathcal{N}\left(\mu_t(s_n^{(0)}), \sigma_t^2(s_n^{(0)})\right),
\]
where \( h_{t,n} = (h_n) \) and \((\cdot)^{0}\) denotes element-wise exponentiation. As represented in Figure 1b, \( \mu_t : \mathbb{R}_c^d \rightarrow \mathbb{R}_c^{L-1} \) and \( \sigma_t^2 : \mathbb{R}_c^d \rightarrow \mathbb{R}_c^{L-1} \) are non-linear functions corresponding to a neural network that takes as input the speech power spectrum at a given time frame. \( \psi \) denotes the parameters of this recognition network, which also have to be estimated by maximizing the variational lower bound defined in (3). Using (1), (2) and (4) we can develop this objective function as follows:
\[
\mathcal{L}(\theta_s, \psi) \overset{\mathcal{E}_s}{=} -\sum_{f=0}^{F-1} \sum_{n=0}^{N-1} \mathcal{L}(s_{f,n} | s_{f,n}^{(0)}; \theta_s, \psi) + \frac{1}{2} \sum_{l=0}^{L-1} \sum_{n=0}^{N-1} \ln \sigma_l^2(s_n^{(2)}) - \mu_l(s_n^{(2)})^2 - \sigma_l^2(s_n^{(2)}),
\]
where \( \mathcal{L}(s_{f,n} | s_{f,n}^{(0)}; \theta_s, \psi) = \mathbb{E}_q[\ln p(s_{f,n} | s_{f,n}^{(0)})] \). Finally, using the so-called "reparametrization trick" [8] to approximate the intractable expectation in (5), we obtain an objective function which is differentiable with respect to both \( \theta_s \) and \( \psi \) and can be optimized using gradient-ascent-based algorithms. It is important to note that the only reason why the recognition network is introduced is to learn the parameters of the generative network.

3. NOISE AND MIXTURE MODELS

In the previous section we have seen how to learn the parameters of the generative model (1)-(2). This model can then be used as a speech signal probabilistic prior for a variety of applications. In this paper we are interested in single-channel speech enhancement. We do not assume prior knowledge about the recording environment, so that the noise model remains unsupervised. Independently for all \( (f, n) \in \mathbb{R} \), the STFT coefficients of noise are modeled as complex circularly symmetric \( \alpha \)-stable random variables [24]:
\[
b_{f,n} \sim \mathcal{S}_{\alpha}  \mathcal{S}(\alpha_{b,f}),
\]
where \( \alpha \in [0, 2] \) is the characteristic exponent and \( \alpha_{b,f} \in \mathbb{R}_+ \) is the scale parameter. As proposed in [20] for multichannel speech enhancement, this scale parameter is only frequency-dependent, it does not depend on time. For algorithmic purposes, the noise model (6) can be conveniently rewritten in an equivalent scale mixture of Gaussians form [25], by making use of the product property of the symmetric \( \alpha \)-stable distribution [24]:
\[
\phi_{f,n} \sim \mathcal{P}_\alpha \mathcal{S}\left(2 \cos(\pi \alpha/4)^{1/2}\right); \quad b_{f,n} | \phi_{f,n} \sim \mathcal{N}(0, \phi_{f,n} \alpha_{b,f}^2),
\]
where \( \phi_{f,n} \in \mathbb{R}_+ \) is called the impulse variable. It locally modifies the variance of the conditional distribution of \( b_{f,n} \) given \( \phi_{f,n} \). \( \mathcal{P}_\alpha \mathcal{S} \) denotes a positive stable distribution of characteristic exponent \( \alpha/2 \). It corresponds to a right-skewed heavy-tailed distribution defined for non-negative random variables [26]. These impulse variables can be understood as carrying uncertainty about the stationary noise assumption made in the marginal model (6), where the scale parameter does not depend on the time-frame index.

The observed mixture signal is modeled as follows for all \( (f, n) \in \mathbb{B} \):
\[
x_{f,n} = \sqrt{g_n} s_{f,n} + b_{f,n},
\]
where \( g_n \in \mathbb{R}_+ \) represents a frame-dependent but frequency-independent gain. The importance of this parameter was experimentally shown in [10]. We further consider the conditional independence of the speech and noise STFT coefficients so that:
\[
x_{f,n} | h_n, \phi_{f,n} \sim \mathcal{N}(0, g_n \alpha_{s,f}^2(h_n) + \phi_{f,n} \alpha_{b,f}^2),
\]

4. INFERENCE

Let \( \theta_s = \{g_n \in \mathbb{R}_+\}_{n=0}^{N-1}, \alpha_{b,f} = \{\alpha_{b,f} \in \mathbb{R}_+\}_{f=0}^{F-1} \) be the set of model parameters to be estimated. For maximum likelihood estimation, in this section we develop a Monte-Carlo expectation maximization (MCEM) algorithm [22], which iteratively applies the so-called E- and M-steps until convergence, which we detail below. Remember that the speech generative model parameters \( \theta_s \) have been learned during a training phase (see Section 2). We denote by \( x = \{x_{f,n}^{(0)}\}_{f=0}^{F-1} \) the set of observed data while \( z = \{h_n, \phi_{f,n} \} \) is the set of latent variables. We will also use \( z_n = \{x_{f,n}^{(0)}\}_{f=0}^{F-1} \) and \( z_n = \{h_n, \phi_{f,n} \} \) to respectively denote the set of observed and latent variables at a given time frame \( n \).

Monte Carlo E-Step. Let \( \theta_s' \) be the current (or the initial) value of the model parameters. At the E-step of a standard expectation-maximization algorithm, we would compute the following conditional expectation of the complete-data log-likelihood:
\[
Q(\theta_s' | \theta_s')_{\mu} = \mathbb{E}_{q(x | z, \theta_s', \theta_s')}[\ln p(x | z; \theta_s, \theta_s')].
\]
However, this expectation cannot be here computed analytically. We therefore approximate \( Q(\theta_s' | \theta_s')_{\mu} \) using an empirical average:
\[
\hat{Q}(\theta_s', \theta_s')_{\mu} \overset{\mu}{=}, \frac{1}{R} \sum_{r=1}^{R} \sum_{(f,n)\in\mathbb{B}} \ln \left(g_n \alpha_{s,f}^2(h_n^{(r)}) + \phi_{f,n} \alpha_{b,f}^2\right) + \left|x_{f,n}^{(r)} - \left(g_n \alpha_{s,f}^2(h_n^{(r)}) + \phi_{f,n} \alpha_{b,f}^2\right)^{-1}\right|^2.
\]
where \( r \) denotes equality up to an additive constant, and \( z_n^{(r)} = \{h_n^{(r)}, \phi_{f,n}^{(r)} \} \) is a sample drawn from the posterior \( p(z_n | x_n; \theta_s', \theta_s') \) using a Markov Chain Monte Carlo (MCMC) method. This approach forms the basis of the MCEM algorithm [22]. Note that unlike the standard EM algorithm, it does
not ensure an improvement in the likelihood at each iteration. Nevertheless, some convergence results in terms of stationary points of the likelihood can be obtained under suitable conditions [27].

In this work we use a (block) Gibbs sampling algorithm [28]. From an initialization \( z_0^{(0)} \), it consists in iteratively sampling from the so-called full conditionals. More precisely, at the \( m \)-th iteration of the algorithm and independently for all \( n \in \{0, \ldots, N-1\} \), we first sample \( h_n^{(m)} \sim p(h_n | x, \phi^{(m-1)}; \theta, \theta^*) \). Then, we sample \( \phi_f^{(m)} \sim p(\phi_f | x, f, h_n^{(m)}; \theta, \theta) \) independently for all \( f \in \{0, \ldots, F-1\} \). Those two full conditionals are unfortunately analytically intractable, but we can use one iteration of the Metropolis-Hastings algorithm to sample from them. This approach corresponds to the Metropolis-within-Gibbs sampling algorithm [28, p. 393]. One iteration of this method is detailed in Algorithm 1. The proposal distributions for \( h_n \) and \( \phi_f \) are respectively given in lines 2 and 6. The two acceptance probabilities required in lines 3 and 7 are computed as follows:

\[
\alpha_n^{(h)} = \min \left( 1, \frac{p(\tilde{h}_n) \prod_{f=0}^{F-1} \prod_{j \in I_f} p(x_{f,j} | \tilde{h}_n, \phi_f^{(m-n)}; \theta, \theta^*)}{p(h_n^{(m-n)}) \prod_{f=0}^{F-1} \prod_{j \in I_f} p(x_{f,j} | h_n^{(m-n)}, \phi_f^{(m-n)}; \theta, \theta^*)} \right)
\]
\[
\alpha_n^{(\phi)} = \min \left( 1, \frac{p(x| h_n^{(m)}; \phi_f; \theta, \theta^*) p(\tilde{\phi}_f) \prod_{f=0}^{F-1} \prod_{j \in I_f} p(x_{f,j} | h_n^{(m)}, \phi_f^{(m-n)}; \theta, \theta^*)}{p(x| h_n^{(m-n)}; \phi_f; \theta, \theta^*) \prod_{f=0}^{F-1} \prod_{j \in I_f} p(x_{f,j} | h_n^{(m-n)}, \phi_f^{(m-n)}; \theta, \theta^*)} \right)
\]

The two distributions involved in the computation of those acceptance probabilities are defined in (1) and (10). Finally, we only keep the last \( R \) samples for computing \( \hat{Q}(\theta, \theta^*) \), i.e. we discard the samples drawn during a so-called burn-in period.

M-Step. At the M-step we want to minimize \(-\hat{Q}(\theta, \theta^*)\) with respect to \( \theta \). Let \( C(\theta) \) denote the cost function associated with this non-convex optimization problem. Similar to [29], we adopt a majorization-minimization approach. Let us introduce the two following sets of auxiliary variables \( c = \{c_f \in \mathbb{R} \}_{f=0}^{F}, f, n \) and \( \lambda = \{\lambda_{f,n} \in \mathbb{R} \}_{k,r,f,n} \). We can show using standard concave/convex inequalities (see e.g. [29]) that \( C(\theta) \leq \mathcal{G}(\theta, c, \lambda) \), where:

\[
\mathcal{G}(\theta, c, \lambda) = \frac{1}{R} \sum_{(f,n)} \sum_k \left[ \ln \epsilon^{(r)}(\lambda_{f,n}) + \frac{1}{c_f^{(r)}} \left( g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2 - \epsilon_f^{(r)} \right) + \frac{|x_{f,n}|^2}{g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2} \right],
\]

Moreover, this upper bound is tight, i.e. \( C(\theta) = \mathcal{G}(\theta, c, \lambda) \), for:

\[
\epsilon_f^{(r)} = g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2;
\]

\[
\lambda_{f,n} = g_c \sigma_{s,f}^2 (h_n^{(r)}) \left( g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2 \right)^{-1};
\]

\[
\lambda_{f,n} = g_c \sigma_{s,f}^2 (h_n^{(r)}) \left( g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2 \right)^{-1}.
\]

Minimizing \( \mathcal{G}(\theta, c, \lambda) \) with respect to the model parameters \( \theta \) is a convex optimization problem. By zeroing the partial derivatives of \( \mathcal{G} \) with respect to each scalar in \( \theta \), we obtain update rules that depend on the auxiliary variables. We then replace these auxiliary variables with the formulas given in (15)-(17), which makes the upper bound \( \mathcal{G} \) tight. This procedure ensures that the cost \( C(\theta) \) decreases [30]. The resulting final updates are given as follows:

\[
\begin{align*}
\sigma_{b,f}^2 &\leftarrow \sigma_{b,f}^2 \left( \sum_{n=0}^{N-1} |x_{f,n}|^2 \sum_{r=1}^{R} \phi_f \sigma_{b,f}^2 \right)^{-1/2} \quad \text{if } \alpha_n^{(h)} > u_n^{(h)} \sim \mathcal{U}([0, 1]) \\
g_n &\leftarrow g_n \left( \sum_{f=0}^{F-1} \sum_{r=1}^{R} \sigma_{s,f}^2 \left( \sum_{n=0}^{N-1} |x_{f,n}|^2 \right)^{-1} \right)^{-1/2} \quad \text{if } \alpha_n^{(\phi)} > u_n^{(\phi)} \sim \mathcal{U}([0, 1])
\end{align*}
\]

where we introduced \( \epsilon^{(r)} = g_c \sigma_{s,f}^2 (h_n^{(r)}) + \phi_f \sigma_{b,f}^2 \) in order to ease the notations. Non-negativity is ensured provided that the parameters are initialized with non-negative values.

Speech reconstruction. Once the unsupervised model parameters \( \theta \) are estimated with the MCEM algorithm, we need to estimate the clean speech signal. For all \((f, n) \in \mathcal{B} \), let \( s_{f,n} = \sqrt{g_n} \tilde{S}_{f,n} \) be the scaled version of the speech STFT coefficients. We estimate these variables according to their posterior mean, given by:

\[
\hat{s}_{f,n} = \mathbb{E}(s_{f,n} | x_{f,n}; \theta, \theta^*) = \mathbb{E}(s_{f,n} | x_{f,n}; \theta, \theta^*) \left[ \mathbb{E}(s_{f,n} | x_{f,n}; \theta, \theta^*) \right] \]
Fig. 2: Speech enhancement results obtained with the proposed method as a function of the characteristic exponent $\alpha$ in the noise model (6). $\alpha = 2.0$ actually corresponds to $\alpha = 1.999$. The value of the median is indicated within each boxplot.

Fig. 3: Results obtained with the reference and the proposed methods (using $\alpha = 1.8$). The median is indicated within each boxplot.

and $H_0$ are estimated from the noisy mixture signal. This method also relies on an MCEM algorithm. By comparing the proposed method with [10], we fairly investigate which noise model leads to the best speech enhancement results. Note that the method proposed in [10] was shown to outperform both a semi-supervised NMF baseline and the fully-supervised deep learning approach [6], the latter having difficulties for generalizing to unseen noise types. We do not include here the results obtained with these two other methods.

**Database:** The supervised speech model parameters in the proposed and the reference methods are learned from the training set of the TIMIT database [31]. It contains almost four hours of 16-kHz speech signals, distributed over 462 speakers. For the evaluation of the speech enhancement algorithms, we artificially added clean speech signals from the TIMIT test set and noise signals from the DEMAND database [32], corresponding to various noisy environments: domestic, nature, office, indoor public spaces, street and transportation. We created 168 mixtures at a 0-dB signal-to-noise ratio (one mixture per speaker in the TIMIT test set). Note that both speakers and sentences are different than in the training set.

**Parameter setting:** The STFT is computed using a 64-ms sine window (i.e., $F = 513$) with 75%-overlap. Based on [10], the latent dimension in the speech generative model (1)-(2) is fixed to $L = 64$. In this reference method, the NMF rank of the noise model is fixed to $K = 10$. The NMF parameters are randomly initialized. For both the proposed and the reference method, the gain $g_n$ is initialized to one for all time frames $n$. For the proposed method, the noise scale parameter $\sigma_{0,f}$ is also initialized to one for all frequency bins. We run 200 iterations of the MCEM algorithm. At each Monte-Carlo E-Step, we run 40 iterations of the Metropolis-within-Gibbs algorithm and we discard the first 30 samples as the burn-in period. The parameter $\varepsilon^2$ in line 2 of Algorithm 1 is set to 0.01.

**Neural network:** The structure of the generative and recognition networks is the same as in [10] and is represented in Fig. 1. Hidden layers use hyperbolic tangent ($\text{tanh}()$) activation functions and output layers use identity activation functions ($I_0()$). The output of these last layers is therefore real-valued, which is the reason why we consider logarithm of variances. For learning the parameters $\Theta$ and $\Phi$, we use the Adam optimizer [33] with a step size of $10^{-3}$, exponential decay rates for the first and second moment estimates of 0.9 and 0.999 respectively, and an epsilon of $10^{-7}$ for preventing division by zero. 20% of the TIMIT training set is kept as a validation set, and early stopping with a patience of 10 epochs is used. Weights are initialized using the uniform initializer described in [34].

**Results:** The estimated speech signal quality is evaluated in terms of standard energy ratios expressed in decibels (dBs) [35]: the signal-to-distortion (SDR), signal-to-interference (SIR) and signal-to-artifact (SAR) ratios. We also consider the perceptual evaluation of speech quality (PESQ) [36] measure (in $[-0.5, 4.5]$), and the short-time objective intelligibility (STOI) measure [37] (in $[0, 1]$). For all measures, the higher the better. We first study the performance of the proposed method according to the choice of the characteristic exponent $\alpha$ in the noise model (6). Results presented in Fig. 2 indicate that according to the PESQ and STOI measures, the best performance is obtained for $\alpha = 2$ (Gaussian case). The SDR indicates that we should choose $\alpha = 1.8$. Indeed, for greater values of $\alpha$, the SIR starts to decrease ($\approx$ 1-dB difference between $\alpha = 1.8$ and $\alpha = 2$) while the SAR remains stable (results are not shown here due to space constraints). Therefore, in Fig. 3 we compare the results obtained with the reference [10] and the proposed method using $\alpha = 1.8$. With the proposed method, the estimated speech signal contains more interferences (SIR is lower) but less artifacts (SAR is higher). According to the SDR both methods are equivalent. But for intelligibility and perceptual quality, artifacts are actually more disturbing than interferences [38], which is the reason why the proposed method obtains better results in terms of both STOI and PESQ measures. For reproducibility, a Python implementation of our algorithm and audio examples are available online. 

6. CONCLUSION

In this work, we proposed a speech enhancement method exploiting a speech model based on VAEs and a noise model based on alpha-stable distributions. At the expense of more interferences, the proposed $\alpha$-stable noise model reduces the amount of artifacts in the estimated speech signal, compared to the use of a Gaussian NMF-based noise model as in [10]. Overall, it is shown that the proposed approach improves the intelligibility and perceptual quality of the enhanced speech signal. Future works include extending the proposed approach to a multi-microphone setting using multivariate $\alpha$-stable distributions [18, 20].

1 Actually $\alpha = 1.999$ because for $\alpha = 2$ the positive $\alpha$-stable distribution in (6) is degenerate.

2 https://team.inria.fr/perception/icassp2019-anvae/
7. REFERENCES


