ON TRACKING NOISE WITH LINEAR DYNAMICAL SYSTEM MODELS

Bhiksha Raj[1], Rita Singh[2] and Richard Stern[2]
1. Mitsubishi Electric Research Labs, Cambridge, MA, USA
2. Department of Electrical and Computer Engineering, Carnegie Mellon University, USA
bhiksha@merl.com, rsingh@cs.cmu.edu, rms@cs.cmu.edu

ABSTRACT

This paper investigates the use of higher-order autoregressive vector predictors for tracking the noise in noisy speech signals. The autoregressive predictors form the state equation of a linear dynamical system that models the spectral dynamics of the noise process. Experiments show that the use of such models to track noise can lead to large gains in recognition performance on speech compensated for the estimated noise. However, predictors of order greater than 1 are not observed to improve the performance beyond that obtained with a first-order predictor. We analyze and explain why this is so.

1. INTRODUCTION

Most typical noises that corrupt speech signals, such as the ambient noises encountered on a busy street, in a bar or in a subway, have a large component that is relatively slow-varying. This is illustrated by Figure 1, which shows the spectrograms for short recordings of two common types of noises. We observe two different trends in these spectrograms: a relatively slowly-varying background, superposed with sudden onsets of events. Given the average trends in these noises, it is reasonable to assume that the noises encountered at any instant of time are a good indicator of what may follow. Hence, one may attempt to predict the future behavior of such noises based on their current and past behavior.

The predictability can be codified by representing the noise as the output of an auto-regressive (AR) process. We note at that outset that our assertions on the predictability of the noise signals relate to the variations of their spectral characteristics and not that of the underlying time-domain noise signal. Also, we choose to ignore phase characteristics of the noise, concentrating primarily on the predictability of the magnitude of spectral magnitudes of the noise. The corresponding statistical model, therefore, models the sequence of *power-spectral vectors* derived from a short-time Fourier transform analysis of the noise signal as a function of the output of an AR process.

In this paper we investigate the use of such a model, for the purpose of compensating for the effects of non-stationary noise on a speech recognition system. We model the sequence of log-spectral vectors of the noise as the output of an AR process. The observed signal is the noisy speech signal - where the speech is the non-linear obscuring influence that prevents us from observing the noise. The combination of the AR equation representing the noise process, and the non-linear equation that relates the noise to the observed noisy speech, form the state and observation equations of a traditional linear dynamical system.

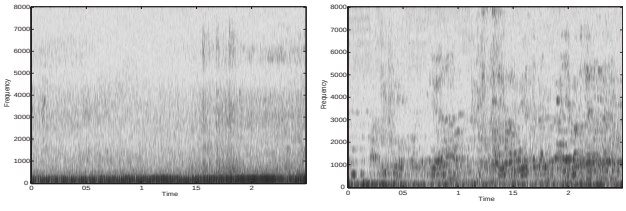

Figure 1: Wide-band spectrogram for a) traffic noise, b) babble

In an earlier paper [1] we presented a simple particle-filtering-based algorithm to estimate the noise spectrum from the observed noisy speech based on such a dynamical system model, where the noise was modeled by a first-order AR process. In this paper we further investigate the applicability and extensibility of the model, including the rationale behind the choice of the feature representation for the noise model, the actual applicability of the AR model to the noises in consideration, and the effect of increasing the AR order on the estimation of the noise spectrum. We note that several aspects of the presented analysis have been discussed earlier by other researchers, in various contexts. The use of dynamical systems to represent noise, in the context of speech recognition dates back to the seminal work of Varga and Moore [2], who represent the noise as the output of a hidden Markov model (HMM). In their work, the HMM-based representation of the dynamics of the process underlying the noise was utilized to improve the performance of a speech recognition system on noisy speech, without explicit compensation of the noisy speech for the noise. Kim *et. al.* [3] have proposed the use of a linear dynamical system to track noise, for explicit compensation of the spectral vectors derived from noisy speech. They use a simplified extended Kalman filter (EKF) formulation for estimating the noise. To render the algorithm stable, they resort to reducing the Kalman gain in a manner that is not mathematically justified. The algorithm itself only permits the use of a simple first-order AR process for the noise. The explicit use of higher-order AR processes to model the noise has not been reported in the speech literature so far, to the best of our knowledge.

The rest of this paper is arranged as follows: In Section 2 we discuss the AR model used to capture the noise dynamics. In Section 3 we describe the dynamical system model used to estimate the noise. In Section 4 we describe the particle-filtering algorithm used to estimate the noise. In Section 5 we describe our experiments, and in Section 6 we discuss our findings.

2. AR MODEL FOR NOISE SPECTRA

The M^{th} order AR model attempts to predict the t^{th} spectral vector for the noise, n_t, as a linear combination of the previous M vectors. Notationally, this can be represented as

$$\hat{n}_t = A_1 n_{t-1} + A_2 n_{t-2} + \ldots + A_M n_{t-M} \quad (1)$$

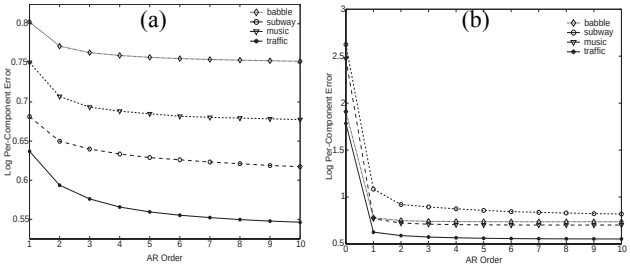

Figure 2: Average per-component prediction error, as a function of AR order (a) on training data, (b) on test data. The right panel also shows the error obtained with a 0th order AR predictor.

where $\hat{n}_t$ is the predicted value for n_t, M is the order of the AR predictor, and the A_i s are the prediction coefficient matrices. The AR model might be defined either for the power-spectral vectors or the log-spectral vectors of the noise, *i.e.* n_t might either represent a power-spectral vector or a log-spectral vector. The prediction coefficient matrices are estimated by minimizing $E[\|\hat{n}_t - n_t\|^2]$, the expected value of the squared norm of the error between the true and predicted value of n_t. For notational simplicity, we define the matrix $A' = [A_1 A_2 \ldots A_M]$. Further we define the extended column vector $\bar{n}_t = [n_t^T n_{t-1}^T \ldots n_{t-M-1}^T]^T$, obtained by vertical concatenation of the vectors $n_t, n_{t-1}, \ldots, n_{t-M-1}$. The AR equation can be rewritten in terms of these terms as $n_t = A'\bar{n}_{t-1}$. Minimization of the expected prediction error norm results in the following estimate for A':

$$A' = E[n_t \bar{n}_{t-1}^T] E[\bar{n}_t \bar{n}_t^T]^{-1} \tag{2}$$

$E[n_t \bar{n}_{t-1}^T]$ and $E[\bar{n}_t \bar{n}_t^T]$ are estimated from a segment of training examples of the noise.

We model the dynamics of log-spectral vectors with the AR model, rather than that of power spectra, since empirical observations show that log-spectral vectors are much better suited for AR models. The average per-component prediction error for log-spectral vectors is much lower than the log of the average per-component prediction error for power spectral vectors.

Increasing the prediction order M usually decreases (and never increases) the average prediction error norm on the training data. Figure 2a shows the average prediction error on the training data, as a function of AR order, for four different noise types. The AR predictors for each of the noises were trained from 60 second long training recordings, which were segmented into 25ms wide non-overlapping frames. Each frame was parameterized into a 32-component Mel-frequency log-spectral vector. AR predictors were estimated both for the sequences of Mel power spectra, and those of the Mel-log-spectra derived from the analysis. Figure 2b shows the prediction error obtained on 30-minute long noise segments not used for training the predictors. The predicted error is seen to decrease monotonically with increasing prediction order, in all cases. The estimated predictors generalize well to data outside of the training data, indicating that the AR model is indeed able to capture general characteristics of the noise, and not merely the trends within the training data.

3. DYNAMICAL SYSTEM FOR NOISE

A dynamical system can be described by two equations: a state equation that specifies the state dynamics of the system, and an observation equation that relates the underlying state of the system to the measurements of the output of the system. For systems with Markovian dynamics, the state equation can be written as

$$s_t = f(s_{t-1}, \varepsilon_t) \tag{3}$$

where s_t, the state at any time t, is a function of the state at time $t-1$ and a driving term ε_t. The state is thus a continuous-valued variable. The output of the system at any time is usually assumed to be dependent only on the state of the system at that time. The observation equation can be represented as

$$o_t = g(s_t, \gamma_t) \tag{4}$$

where o_t is the observation at time t and γ_t represents any noise affecting the system at time t.

We designate the M^{th} order AR predictor for the log-spectral vectors of the noise as the basis of our state equation. In order to facilitate the employment of estimation procedures designed for first-order Markov processes, we restate the M^{th} order predictor of Equation (1) as a first-order regression, in terms of $\bar{n}_t$, where, as before, $\bar{n}_t = [n_t^T n_{t-1}^T \ldots n_{t-M-1}^T]^T$. $\bar{n}_t$ is thus the state of the system. We define the matrix $\bar{A}$ as

$$\bar{A} = \begin{bmatrix} A' \\ I \;\; 0 \end{bmatrix} \tag{5}$$

where I is an $(M-1)D \times (M-1)D$ identity matrix, where D is the dimensionality of the noise log-spectral vector n_t, and $A' = [A_1 A_2 \ldots A_M]$, as before. Equation (3) can be written as

$$\bar{n}_t = \bar{A}\bar{n}_{t-1} + \varepsilon_t \tag{6}$$

Traditionally, the ε_t terms are assumed to be samples from a 0 mean Gaussian random process. However, in Equation (6) the lower $(M-1)D$ components of the state vector at time t, $\bar{n}_t$, are identically equal to the first $(M-1)D$ components of $\bar{n}_{t-1}$. As a result, the prediction error for the last $(M-1)D$ components of $\bar{n}_t$, *i.e.* the last $(M-1)D$ components of ε_t, is always equal to 0, and only the subvector formed from the first D components of ε_t is assumed to have a Gaussian distribution, whose covariance φ_ε is learned from training data. The mean of ε_t is assumed to be 0.

The observation equation for the dynamical system is the relationship between y_t, the noisy speech log-spectral vectors, $\bar{n}_t$, the state of the dynamical system, and x_t, the hypothetical log-spectral vector for clean speech that would have been observed had it not been corrupted by the noise. This relation is given by the following equation [4]:

$$y_t = f(x_t, \bar{n}_t) = x_t + \log(1 + e^{B\bar{n}_t - x_t}) \tag{7}$$

where $B = [I, 0]$, where I is a D-dimensional identity vector. Equations (6) and (7) thus represent the final state and observation equations respectively.

4. PARTICLE FILTERING ALGORITHM TO ESTIMATE NOISE

The problem we address next is that of determining the state of the system, namely the noise n_t, given only the sequence of

observations y_t, the parameters of the state equation, $\bar{A}$, and φ_ε, and the distribution of x_t. We model the distribution of x_t by a mixture Gaussian density of the form

$$P(x_t) = \sum_{k=1}^{K} c_k N(x_t; \mu_k, \sigma_k) \quad (8)$$

where c_k, μ_k and σ_k represent the mixture weight, mean and variance respectively of the k^{th} Gaussian, and $N(x_t; \mu_k, \sigma_k)$ represents a Gaussian with mean μ_k and variance σ_k.

Defining the prediction and update equations: For ease of presentation we introduce the following notation: we represent the sequence of observations $y_0, y_1, \ldots, y_t$ as $y_{0,t}$. The *a posteriori* probability distribution of the state of the system at time t, given the sequence of observations $y_{0,t}$ can be obtained through the following recursion:

$$P(\bar{n}_t | y_{0,t}) = CP(\bar{n}_t | y_{0,t-1}) P(y_t | \bar{n}_t) \quad (9)$$

$$P(\bar{n}_t | y_{0,t-1}) = \int_{-\infty}^{\infty} P(\bar{n}_t | \bar{n}_{t-1}) P(\bar{n}_{t-1} | y_{0,t-1}) d\bar{n}_{t-1} \quad (10)$$

where C is a normalizing constant. Equation (9), also referred to as the *update* equation requires the computation of $P(y_t | \bar{n}_t)$. Since x_t, the clean speech vector at any time t may have been generated by any of the K Gaussians in the Gaussian mixture distribution in Equation (8) with probability c_k, we get

$$P(y_t | \bar{n}_t) = \sum_{k=1}^{K} c_k P(y_t | \bar{n}_t, k) \quad (11)$$

where $P(y_t | \bar{n}_t, k)$ is the probability of y_t, conditioned on $\bar{n}_t$, and given that the clean speech vector x_t was generated by the k^{th} Gaussian in the mixture. From Equation (7), we can derive the following value for $P(y_t | \bar{n}_t, k)$:

$$P(y_t | \bar{n}_t, k) = \frac{N(y_t + \log(1 - e^{B\bar{n}_t - y_t}); \mu_k, \sigma_k)}{\left| 1 - e^{B\bar{n}_t - y_t} \right|} \quad \text{if } y_t \geq B\bar{n}_t \quad (12)$$

Equations (9), (11) and (12) together define the *update* equation. Equation (12) cannot be used directly in the Kalman recursion Equations (9) and (10), as it results in non-closed-form solutions. Both Kim *et. al.* [3], and our earlier paper [1] approximate Equation (7) with a linear equation derived from a Taylor series expansion, in order to reduce $P(y_t | \bar{n}_t, k)$ to a more tractable Gaussian density. However, the particle-filtering based algorithm presented here does not require this approximation.

Equation (10), the *prediction* equation, requires the computation of $P(\bar{n}_t | \bar{n}_{t-1})$, which is given by

$$P(\bar{n}_t | \bar{n}_{t-1}) = N(n_t; 0, \varphi_\varepsilon) \delta(\bar{n}_t^- - \bar{n}_{t-1}^+) \quad (13)$$

where φ_ε is the covariance matrix for ε_t, n_t refers to the vector obtained from the first D components of $\bar{n}_t$, $\bar{n}_t^-$ refers to the vector of the last $(M-1)D$ components of $\bar{n}_t$, and $\bar{n}_{t-1}^+$ refers to the vector of the first $(M-1)D$ components of $\bar{n}_{t-1}$. Since $P(\bar{n}_t | \bar{n}_{t-1})$ is Gaussian for the first D components of $\bar{n}_t$, and effectively a Dirac delta for the other components, direct computation of Equation (10) results in a peculiar solution where the distribution of the last $(M-1)D$ dimensions of $\bar{n}_t$ as given by $P(\bar{n}_t | y_{0,t-1})$ is identical to that of the first $(M-1)D$ components of $\bar{n}_{t-1}$, as given by $P(\bar{n}_{t-1} | y_{0,t-1})$. This, however, is not a complication for the particle-filtering algorithm we use.

The particle filtering algorithm: The particle filter algorithm is a sampling based algorithm that discretizes the predicted noise distribution at any instant by redefining it as a uniform distribution over a discrete set of samples drawn from the original predicted distribution [5]. Procedurally, at each time instant t, a set of N samples are drawn from the predicted continuous density $P(\bar{n}_t | y_{0,t-1})$. The predicted density is then approximated by a uniform discrete distribution over these generated samples as:

$$P(\bar{n}_t | y_{0,t-1}) \approx \frac{1}{N} \sum_{k=0}^{N-1} \delta(\bar{n}_t - \bar{n}_t^k) \quad (14)$$

where $\bar{n}_t^k$ is the k^{th} noise sample generated from the continuous density $P(\bar{n}_t | y_{0,t-1})$, and N is the total number of samples generated from it. Thereafter, the update equation simply becomes

$$P(\bar{n}_t | y_{0,t}) = C \sum_{k=0}^{N-1} P(y_t | \bar{n}^k) \delta(\bar{n}_t - \bar{n}_t^k) \quad (15)$$

where C is a normalizing constant that ensures that the total probability sums to 1.0. $P(y_t | \bar{n}_t^k)$ is computed using Equation (11). The prediction equation for time $t+1$ now becomes:

$$P(\bar{n}_{t+1} | y_{0,t}) = C \sum_{k=0}^{N-1} P(y_t | \bar{n}_t^k) P(\bar{n}_{t+1} | \bar{n}_t^k) \quad (16)$$

This is a mixture of N distributions of the form $P(\bar{n}_{t+1} | \bar{n}_t^k)$, where $P(\bar{n}_{t+1} | \bar{n}_t^k)$ is computed using Equation (13). This is once again sampled to approximate it as in Equation (14).

To initialize the recursion with $P(\bar{n}_0 | y_{0,-1})$, we draw samples from $P(n_t)$, the *a priori* distribution of n_t, and duplicating each sample M times to create an MD dimensional sample as follows:

$$(\bar{n}_0^k)^T = [(n^k)^T, (n^k)^T, (n^k)^T, \ldots]^T \quad (17)$$

where n^k is the k^{th} sample drawn from $P(n_t)$, and T refers to the transposition operation.

Compensating for the noise: For each frame of incoming noisy speech, the algorithm described above estimates a discrete *a posteriori* distribution of the form:

$$P(\bar{n}_t | y_{0,t}) = C \sum_{k=0}^{N-1} P(y_t | \bar{n}^k) \delta(\bar{n}_t - \bar{n}_t^k) \quad (18)$$

For any estimate of the noise, $\bar{n}_t^k$, we can estimate x_t, the log spectrum of the clean speech, from y_t the log spectrum of the observed noisy speech, using the approximated minimum mean squared estimation (MMSE) procedure developed in [4] as:

$$\hat{x}_t^k = y_t - \sum_{j=1}^{K} p(j|y_t, \bar{n}_t^k)\log\left(1 + e^{B\bar{n}_t^k - \mu_j}\right) \quad (19)$$

where $p(j|y_t, \bar{n}_t^k)$ is given by

$$p(j|y_t, \bar{n}_t^k) = \frac{c_j P(y_t|\bar{n}_t^k, j)}{\sum_{i=1}^{K} c_i P(y_t|\bar{n}_t^k, i)} \quad (20)$$

Combining Equations (18) and (19) gives the estimate for x_t as

$$\hat{x}_t = y_t - C\sum_{k=0}^{N-1} P(y_t|\bar{n}_t^k)\sum_{j=1}^{K} p(j|y_t, \bar{n}_t^k)\log\left(1 + e^{B\bar{n}_t^k - \mu_j}\right) \quad (21)$$

5. EXPERIMENTS

We investigated the effectiveness of the AR noise model by testing recognition performance on noise-corrupted speech compensated by the algorithm described in Section 4. The database used was a Spanish telephone speech database provided by Telefonicá Investigación y Desarrollo (TID). The CMU Sphinx-3 speech recognition system was used in the experiments. Continuous density 8 Gaussian/state HMMs with 500 tied states were trained from 3500 utterances of clean telephone recordings. The test data consisted of telephone recordings corrupted to various SNRs by traffic noise, music, babble, and noise recordings from a subway.

AR models of various orders were estimated from 30-second training recordings of each of the noises. The predicted state (noise) distributions were discretized by drawing 25 samples from them. Clean speech log spectra were estimated from the log spectra of the noisy speech using the MMSE procedure in Section 4. Cepstra derived from the estimated clean speech log spectra were used for recognition.

Figure 3 shows recognition results obtained for the various noise types as a function of SNR, using AR models of orders 1, 2 and 3. As a comparison, recognition with uncompensated noisy speech, and with cepstra derived by VTS compensation [4] are also shown. The VTS algorithm assumes stationary noise, which is equivalent to considering an AR model of order 0 for then noise, albeit without the sampling-based approach.

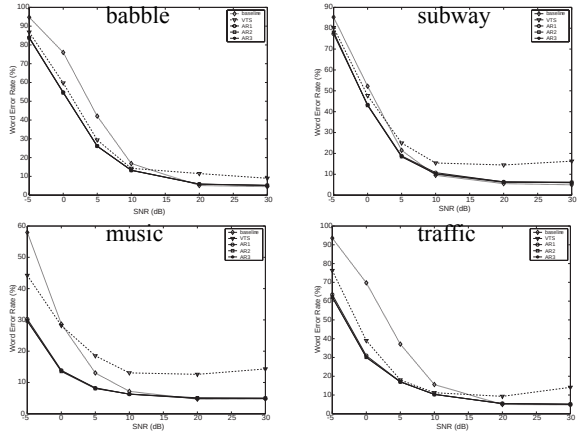

Figure 3: Recognition performance on speech corrupted by four different types of noises. Baseline word error rate (WER), and WERs obtained VTS and the proposed dynamical systems algorithm are all shown.

6. DISCUSSION

The figures in Section 2 show that the AR model can effectively describe the spectral dynamics of several kinds of noises typically encountered during speech recognition. The experiments reported in Section 5 confirm this to some degree. We observe from Figure 3 that the AR model based algorithms are generally highly effective at improving recognition performance on speech corrupted to low SNRs by noise. At higher SNR the 0th order AR model is ineffective in most cases, presumably since the noises used are in reality non-stationary, whereas the 0th order model assumes stationary noise. The first order AR model is able to compensate well for all noises at almost all SNRs, and is significantly better than the 0th order AR model. However, increasing the AR order any further does not result in any additional improvement in the recognition accuracy, although the higher order AR model itself is better able to predict the noise process, as shown by Figure 2. The reason for the absence of improvement in recognition performance, with increasing AR order may be intuited from Figure 2b. The 0th order predictor in Figure 2b simply predicts every spectral vector as the mean of the *a priori* distribution. The largest improvement in prediction error is observed when the AR order increases from 0 to 1. Further increase in the AR order results only in relatively miniscule improvements in prediction error. This correlates well with the results in Section 5, and suggests that the prediction error of any predictive model for corrupting noise may be used as an indicator of the potential gains in recognition accuracy that might be obtained with that model.

Finally, Equation (12) is derived assuming that the power spectral value of the noisy speech is never lesser than that of the noise itself. This assumption, in turn, is based on the assumption that the corrupting noise is perfectly uncorrelated with the speech signal itself. In practice, although the noise is uncorrelated with the speech in the long term, within any given analysis window of finite length it is possible for the energy in the noise spectrum to be greater than that of the observed noisy speech. As a result, noise estimates based on Equation (12) tend to be biased. This has however not been observed to affect the performance of the particle-filtering algorithm adversely, in practice.

REFERENCES

[1] Singh, R. and Raj, B. (2003), "Tracking noise via dynamical systems with a continuum of states," Proc. ICASSP 2003.

[2] Varga, A.P. and Moore, R.K. (1990), "Hidden Markov model decomposition of speech and noise," Proc. ICASSP 1990, pp 845-548

[3] Kim, N.S. (1998), "IMM-based estimation for slowly evolving environments," *IEEE Signal Processing Letters*, 5(6), 146-149

[4] Moreno, P.J. (1996), *Speech Recognition in Noisy Environments*, Ph.D Thesis, ECE Department, Carnegie Mellon University

[5] Gordon, N., Salmond, D. and Smith, A. (1993), "A novel approach to non-linear and non-Gaussian Bayesian state estimation," IEE Proceedings-F, vol. 140, pp. 107-113