TRACKING NOISE VIA DYNAMICAL SYSTEMS WITH A CONTINUUM OF STATES

Rita Singh[1] and Bhiksha Raj[2]
1. School of Computer Science, Carnegie Mellon University, USA
2. Mitsubishi Electric Research Labs, USA
rsingh@cs.cmu.edu, bhiksha@merl.com

ABSTRACT

In this paper we model noise as a sequence of states of a dynamical system with a continuum of states. Observations generated by such a system are assumed to be related to the state of the system by a functional relation which models clean speech as the corrupting influence on noise. We show how the closed-form representation of such a dynamical system can be rendered tractable and solved iteratively by dynamically sampling the state space, resulting in an estimated noise sequence (sequence of states), which can then be removed from the noisy speech signal by standard methods. Experiments on speech corrupted by various noises show that the proposed algorithm performs better than our best previous algorithm, VTS, which assumes that the noise is stationary.

1. INTRODUCTION

Any noise sequence is the output of some underlying process. We may not fully know the nature or the parameters of the process. To counter our ignorance, we model the process largely as random, with additional formulaic representation of what little we do know about the system. Dynamical systems represent convenient tools to facilitate such representations. They can accommodate arbitrarily complex processes, diverse sources of information, and are amenable to standard analytical tools when simplified to suitable forms.

In this paper we attempt to represent the underlying process behind any noise using a simple dynamical system with a continuum of states. Using this model, we attempt to track the noise affecting a speech signal.

The conventional approach to estimating noise affecting a speech signal is to model the speech signal as the output of a dynamical system, such as an HMM, and to estimate the noise based on variations of the measured speech signal from typical output of the known underlying system. We, however, treat the problem inversely and assume that it is the *speech* signal that corrupts our observations of the noise. The measurements of the observed speech-corrupted noise are non-linearly related to both, the hypothetical measurements of the noise that would have been made, had there been no corrupting speech, and the corresponding measurement of the corrupting speech in the absence of noise. Note that this is different from the statement that the noise and the corrupting speech are non-linearly combined. Based on this model, we attempt to estimate the noise from its speech-corrupted measurements.

Once the noise is estimated, however, we revert to the conventional approach and attempt to eliminate the estimated noise from the noisy speech signal, assuming that speech is the primary signal to be measured.

In this paper we will call dynamical systems which have a continuum of states as *continuous-state* dynamical systems. While these can be arbitrarily complex, we choose to work with simple systems with linear Markovian dynamics. These represent a first-order fit to any true underlying dynamical system, however complex, and often capture most of the salient features of the underlying system. Also, first-order parameters are fewer and can be robustly learned from a small amount of training data. This is of immense practical value in most situations encountered in speech recognition, wherein noise must be compensated for.

Tracking dynamical systems in an analytical manner becomes difficult when the conditional densities of the output of the system are mixtures of many component densities. This is unfortunately the case in most real-world processes, including noise and speech. In these cases the complexity of the estimated distribution for the state of the system, as measured by the number of parameters in it, increases exponentially with the progression of time. Additionally, when the relationship between the measured output and the true output of the system is non-linear, the estimated state distributions may not have a closed form at all.

In continuous-state dynamical systems such as the ones used in this paper we encounter both these problems. We restrain the complexity of the estimated distribution by sampling predicted distributions for the output of the system at each time step, and propagating these thus discretized distributions to further steps of algorithm. This approach has been successfully used in several problems (e.g. [1,2]). The non-linearity of the relation between the state of the system and the observations is dealt with by linearization using Taylor series expansions around previous estimates of the system state [3].

Section 2 of this paper describes the continuous-state dynamical system used to model noise. Section 3 describes the algorithm used to estimate the noise. Section 4 describes how we estimate clean speech feature vectors from the feature vectors of noisy speech and the estimated distribution of noise. Sections 5 and 6 present an experimental evaluation of the proposed algorithm and related conclusions, respectively.

2. 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 sys-

tem to the measurements of the output of the system. When the state dynamics of the system are assumed to be Markovian, the state equation can be represented as

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

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 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

$$o_t = g(s_t, \gamma_t) \quad (2)$$

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

In many cases, the best set of state and observation equations required to model a system accurately may be quite complex, making the estimation of the state from the observations intractable. In addition, the estimation of the parameters of such a system may be very difficult from finite amounts of data. For these reasons, it is often advantageous to approximate the dynamics with a simple first-order system. In keeping with this argument, we model the dynamics of the system whose states are log-spectrcal vectors of noise as

$$n_t = An_{t-1} + \varepsilon_t \quad (3)$$

where n_t represents the noise log-spectral vector at time t. This is an auto-regressive model of order 1 that assumes that the sequence of noise log-spectral vectors can be modelled as the output of a first-order auto-regressive (AR) system excited by a 0 mean Gaussian process. A represents the AR parameter, and ε_t represents the Gaussian excitation process. The AR parameter A, and the variance of ε_t, φ_ε, can all be learned from a small amount of representative noise samples. The mean of ε_t is assumed to be 0.

The log-spectral vectors of the noisy observations y_t are related to the state of the dynamical system, as represented by n_t, and the log-spectra of the corrupting clean speech x_t by the following equation [3]:

$$y_t = f(x_t, n_t) = x_t + \log(1 + \exp(n_t - x_t)) = x_t + l(x_t, n_t) \quad (4)$$

Equations (3) and (4) represent the state and observation equations respectively. Having thus formulated the dynamical system, 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, 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

$$P(x_t) = \sum_{k=1}^{K} c_k N(x_t; \mu_k, \sigma_k) \quad (5)$$

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.

3. NOISE ESTIMATION ALGORITHM

For ease of presentation we introduce the following notation: we represent the sequence of observations $y_0, y_1, \ldots, y_t$ as $y_{0,t}$. It can easily be shown that 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:

$$P(n_t|y_{0,t-1}) = \int_{-\infty}^{\infty} P(n_t|n_{t-1})P(n_{t-1}|y_{0,t-1})dn_{t-1} \quad (6)$$

$$P(n_t|y_{0,t}) = CP(n_t|y_{0,t-1})P(y_t|n_t) \quad (7)$$

where C is a normalizing constant. Equation (6) is referred to as the *prediction* equation and Equation (7) as the *update* equation. $P(n_t|y_{0,t-1})$ is the predicted distribution for n_t and $P(n_t|y_{0,t})$ is the updated distribution for n_t. The goal of the recursion is to estimate the updated distribution. In this paper we refer to recursions of Equation (6) and Equation (7) as the *Kalman recursion*.

From Equation (3), since ε_t has a Gaussian distribution with mean 0 and variance φ_ε, we obtain the conditional density of n_t, given n_{t-1} as

$$P(n_t|n_{t-1}) = N(n_t; An_{t-1}, \varphi_\varepsilon) \quad (8)$$

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 (5), with probability c_k. We can therefore write:

$$P(y_t|n_t) = \sum_{k=1}^{K} c_k P(y_t|n_t, k) \quad (9)$$

where $P(y_t|n_t, k)$ is the probability of y_t, conditioned on n_t, and given that the clean speech vector x_t was generated by the k^{th} Gaussian in the mixture. It can be shown that [4]:

$$P(y_t|n_t, k) = \frac{N(f^{-1}(y_t, n_t); \mu_k, \sigma_k)}{\left|\frac{dy_t}{dx_t}\right|} \quad (10)$$

where $f^{-1}(y_t, n_t)$ is the inverse function that derives x_t as a function of y_t and n_t, and the Jacobian of y_t in the denominator is the determinant of the derivative of y_t with respect to x_t.

Both $f^{-1}(y_t, n_t)$ and the Jacobian are highly non-linear functions, as a result of which $P(y_t|n_t, k)$ has a form that leads to complicated solutions. In order to avoid this complication, we approximate Equation (4) by a truncated Taylor series, expanded around the mean of the k^{th} Gaussian:

$$l(x_t, n_t) = l(\mu_k, n_t) + l'(\mu_k, n_t)(x_t - \mu_k) + \ldots \quad (11)$$

Higher order terms are not shown in the Equation (11). We truncate this series after the first term, to obtain

$$l(x_t, n_t) \approx l(\mu_k, n_t) \quad (12)$$

This can be used to derive $P(y_t|n_t, k)$ as

$$P(y_t|n_t, k) = N(y_t; \mu_k + l(\mu_k, n_t), \sigma_k) = N(y_t; f(\mu_k, n_t), \sigma_k) \quad (13)$$

We could also truncate the series expansion in Equation (11) after the first order term, and $P(y_t|n_t, k)$ would still be Gaussian. Inclusion of higher order terms in the approximation will, how-

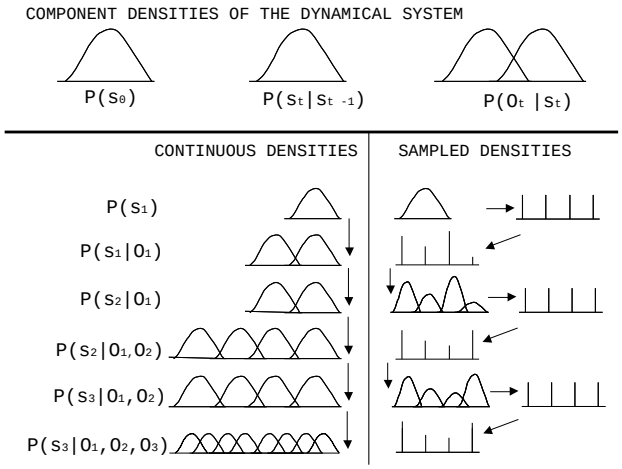

Figure 1. Evolution of densities with the progression of the algorithm. In this example, the *a priori* distribution of the state, and the conditional density of the state at time t given the state at $t-1$, are both Gaussian. The output density, given the state, is a mixture of two Gaussians. In conventional estimation of continuous densities, the updated density at each instant has twice as many Gaussian components as at the previous instant. In the sampling based algorithm, the updated distribution has only as many values as the number of samples derived from the predicted density. This number is entirely controlled by the sampling and does not increase with time.

ever, result in more complicated distributions for $P(y_t|n_t, k)$.

It is important to note that the approximation in Equation (12) is specific to the k^{th} Gaussian. Combining Equation (13) with Equation (9), we get the following approximation for $P(y_t|n_t)$:

$$P(y_t|n_t) = \sum_{k=1}^{K} c_k N(y_t; f(\mu_k, n_t), \sigma_k) \tag{14}$$

The Kalman recursion is initialized using the *a priori* distribution of the noise:

$$P(n_0|y_{0,-1}) = P(n_0) \tag{15}$$

While it is possible to now run the Kalman recursion by direct computations of Equations (6) and (7), it can easily be shown that this results in an exponential increase in the complexity of the updated distribution for n_t with increasing t. In general, the estimated distribution of n_t will be a mixture of K^{t+1} Gaussians. Figure 1 illustrates this problem. The problem could be simplified by collapsing the Gaussian mixture distribution for $P(n_t|y_{0,t})$ into a single Gaussian at every step. However this frequently leads to unsatisfactory solutions and poor tracking of the noise. Instead, we use sampling methods to reduce the problem.

3.1 Sampling the Predicted State Density

The complexity of the *a posteriori* noise distribution can be controlled by discretizing the predicted noise density at each time step. The predicted noise density is sampled to generate a number of noise samples. The continuous density is then replaced by a uniform discrete distribution over these generated samples:

$$P(n_t|y_{0,t-1}) \approx \frac{1}{N}\sum_{k=0}^{N-1} \delta(n_t - n^k) \tag{16}$$

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

$$P(n_t|y_{0,t}) = C\sum_{k=0}^{N-1} P(y_t|n^k)\delta(n_t - n^k) \tag{17}$$

where C is a normalizing constant that ensures that the total probability sums to 1.0. $P(y_t|n^k)$ is computed using Equation (14). The prediction equation for time $t+1$ now becomes:

$$P(n_{t+1}|y_{0,t}) = C\sum_{k=0}^{N-1} P(y_t|n^k)P(n_{t+1}|n^k) \tag{18}$$

This is a mixture of N distributions of the form $P(n_{t+1}|n^k)$. This is once again sampled to approximate it as in Equation (16). The overall algorithm can be summarized as:

1. Set $P(n_0|y_{0,-1}) = P(n_0)$. Set $t = 0$.
2. Generate N samples of noise from $P(n_t|y_{0,t-1})$.
3. Compute $P(n_t|y_{0,t})$ using Equation (17).
4. Compute $P(n_{t+1}|y_{0,t})$ using Equation (18).
5. Set $t = t+1$ and return to step 2.

4. COMPENSATING FOR THE NOISE

The noise estimation algorithm described in Section 3.1 estimates, for each frame of incoming noisy speech, a discrete *a posteriori* distribution of the form:

$$P(n_t|y_{0,t}) = C\sum_{k=0}^{N-1} P(y_t|n^k)\delta(n_t - n^k) \tag{19}$$

For any estimate of the noise, n^k, we 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 [3] as:

$$\hat{x}_t^k = y_t - \sum_{j=1}^{K} p(j|y_t, n^k) f(\mu_j, n^k) \tag{20}$$

where $p(j|y_t, n^k)$ is given by

$$p(j|y_t, n^k) = \frac{c_j N(y_t; f(\mu_j, n^k), \sigma_j)}{\sum_{i=1}^{K} c_i N(y_t; f(\mu_i, n^k), \sigma_i)} \tag{21}$$

Combining Equations (19) and (20), we get the overall estimate for x_t as

$$\hat{x}_t = y_t - C\sum_{k=0}^{N-1} P(y_t|n^k)\sum_{j=1}^{K} p(j|y_t, n^k) f(\mu_j, n^k) \tag{22}$$

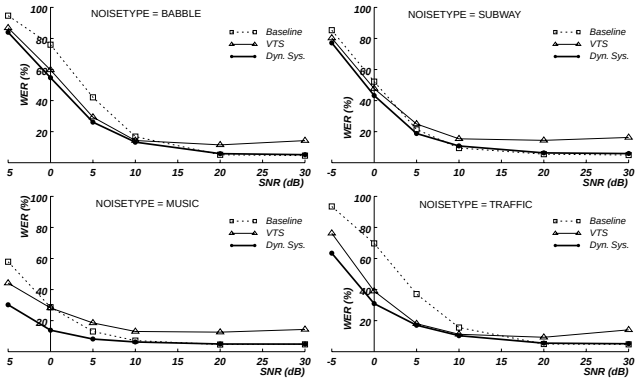

Figure 2. Recognition performance on telephone speech corrupted by four different types of noises. Word error rates (WERs) obtained with noisy speech, compensation with the VTS algorithm, and compensation using the proposed dynamical systems (DS) algorithm are all shown.

5. EXPERIMENTS

The proposed algorithm was evaluated on a Spanish telephone speech database provided by Telefonicá Investigación y Desarrollo (TID) using the CMU Sphinx-3 speech recognition system. 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 recorded in a bar, and noise recordings from a subway. The AR matrix for each noise condition was trained from a training example of the noise. 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 2 shows recognition results obtained for the various noise types as a function of SNR. As a comparison, recognition with uncompensated noisy speech, and with cepstra derived by VTS compensation are also shown. The VTS algorithm has previously been shown to be highly effective at compensating for stationary noises [3]. We observe from Figure 2 that both VTS and the proposed algorithm are highly effective at improving recognition performance at low SNRs. At these SNRs it is apparently advantageous to eliminate even an average characteristic of the noise, regardless of the non-stationary nature of the noise. However, at higher SNRs the VTS algorithm begins to falter, since the noises are all non-stationary. At these SNRs recognition performance with VTS-compensated speech is actually poorer than that obtained with the uncompensated noisy speech. However, the proposed algorithm is able to cope with the nonstationarity of the noise at all SNRs, and performs consistently better than the VTS algorithm. At high SNRs, where the VTS algorithm fails, it continues to provide improvements over recognition with noisy speech. Even at SNRs higher than 20dB, where the speech is essentially clean, the algorithm does not degrade performance to a perceptible degree.

6. CONCLUSIONS

The proposed algorithm uses more information about the noise signal than the VTS algorithm, or other algorithms that assume that the noise is stationary. The amount of explicit information required about the noise is however small, due to the simple first order model assumed for the dynamics. Even this small amount of information permits us to track the noise well. In the format of the algorithm reported in this paper, the type of noise corrupting the speech signal was assumed to be known. In a more generic case, this may not be known. In such situations, one solution would be to have several different dynamical systems trained on a variety of noise types. The most appropriate model for the noise type affecting the signal could then be identified using system or model identification methods [5].

The speech log-spectra are modelled as the output of an IID process, in this paper. They can also be modelled by an HMM, without any significant modification of the algorithm. As an extension we could treat the systems generating the speech and the noise as coupled dynamical systems, and the algorithm can be appropriately modified to simultaneously track both speech and noise.

The dynamical system modelling the noise may itself also be extended. For example, the AR order for the dynamical system has been assumed to be one in this paper. This can easily be extended to higher orders. Additionally, the dynamical system may be made non-linear without major modifications to the algorithm. However, this would require appropriate techniques to learn the parameters of the non-linear dynamical system.

Finally, we note that the proposed algorithm is designed to be an on-line algorithm, as opposed to previously reported algorithms like VTS, which are essentially off-line algorithms that require many passes over the noisy data. The proposed algorithm estimates the noise at each instant without reference to future data enabling the compensation of data as they are encountered.

ACKNOWLEDGEMENTS

Rita Singh was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

REFERENCES

[1] Isard, M. and Blake, E. (1998), CONDENSATION: Conditional density propagation for visual tracking, *Intl. Journal of Computer Vision*, Vol. 29, pp. 5-28.

[2] Carpenter, J., Clifford, P. and Fearnhead, P. (1999), Building robust simulation-based filters for evolving data sets, T*echnical report*, University of Oxford, Dept. of Statistics.

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

[4] Papoulis, A. (2001), *Probability, Random Variables and Stochastic Processes*, McGraw-Hill.

[5] Ljung, L. (1999), *System Identification: theory for the user*, Prentice Hall.