Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9, 2013

Noisy Hidden Markov Models for Speech Recognition

Kartik Audhkhasi, Osonde Osoba, Bart Kosko

***Abstract*— We show that noise can speed training in hidden Markov models (HMMs). The new Noisy Expectation-Maximization (NEM) algorithm shows how to inject noise when learning the maximum-likelihood estimate of the HMM parameters because the underlying Baum-Welch training algorithm is a special case of the Expectation-Maximization (EM) algorithm. The NEM theorem gives a sufficient condition for such an average noise boost. The condition is a simple quadratic constraint on the noise when the HMM uses a Gaussian mixture model at each state. Simulations show that a noisy HMM converges faster than a noiseless HMM on the TIMIT data set.**

***Index Terms*— Hidden Markov model, Expectation Maximization algorithm, noisy EM algorithm, stochastic resonance, speech recognition, noise injection**

I. Noise Benefits in Speech Recognition

We show that careful noise injection can speed the training process for a hidden Markov model (HMM). The proper noise appears to help the training process explore less probable regions of the parameter space. We call the new system a noisy HMM or NHMM. Figure 1 describes the NHMM architecture based on a noise-enhanced version of the expectation-maximization (EM) algorithm. Figure 2 shows that noise produces a 37% reduction in the number of iterations that it takes to converge to the maximum-likelihood estimate. Figure 3 shows simulation instances where the NHMM converges more quickly than does the standard or noiseless HMM that uses Gaussian mixture models. Figure 4 further shows that the NHMM converges faster than an HMM with simple annealed "blind noise" added to the training data. Such blind noise does not satisfy the key sufficient condition in the noise-enhanced EM algorithm.

The new NHMM is a special case of the recent noisy EM (NEM) model [1], [2]. The underlying NEM theorem states that the noise-enhanced EM algorithm converges faster on average to the maximum-likelihood optimum than does the noiseless EM algorithm if the noise obeys a positivity condition. The condition reduces to a quadratic constraint on the injected noise in the special but important case of a Gaussian mixture model. The NEM algorithm gives rise to the NHMM because the Baum-Welch algorithm that trains the HMM parameters is itself a special case of the EM algorithm [3]. Theorem 1 below states the corresponding sufficient condition for an HMM noise boost. This is a type of "stochastic resonance" effect where a small amount of noise improves the performance of a nonlinear system while too much noise harms the system [4]–[21].

Kartik Audhkhasi, Osonde Osoba, and Bart Kosko are with the Signal and Image Processing Institute, Electrical Engineering Department, University of Southern California, Los Angeles, CA (email: kosko@sipi.usc.edu)

The simulations below confirm the theoretical prediction that proper injection of noise can improve speech recognition. This appears to be the first deliberate use of noise injection in the speech data itself. Earlier efforts [22], [23] used annealed noise to perturb the model parameters and to pick an alignment path between HMM states and the observed speech data. These earlier efforts neither added noise to the speech data nor found any theoretical guarantee of a noise benefit.

An HMM is a popular probabilistic model for time series data. Its many applications include speech recognition [24]–[26], computational biology [22], [27], [28], computer vision [29], [30], wavelet-based signal processing [31], and control theory [32]. HMMs are especially widespread in speech processing and recognition. All popular speech recognition toolkits use HMMs: Hidden Markov Model Toolkit (HTK) [33], Sphinx [34], SONIC [35], RASR [36], Kaldi [37], Attila [38], BYBLOS [39], and Watson [40].

HMMs relate to neural networks in several ways. The forward algorithm of Baum-Welch HMM training resembles the training of some recurrent neural networks [41]. Modern automatic speech recognition also relies heavily on both HMMs and neural networks. Neural-HMM hybrid architectures have improved the performance of speech recognition in many cases [42]–[46].

The next section reviews HMMs and the Baum-Welch algorithm that tunes them. Section III reviews the NEM algorithm and Section IV presents the sufficient condition for a noise boost in HMMs. Section V tests the new NHMM algorithm for training monophone models on the TIMIT corpus.

II. Hidden Markov Models

HMMs [24] are probabilistic latent variable models for multivariate time series data. An HMM consists of a time-homogeneous Markov chain with M states and a single-step transition matrix $\mathbf{A}$. Let $S : \mathbb{Z}^+ \rightarrow \mathbb{Z}_M$ denote a function that maps time to state indices. Then

$$\mathbf{A}_{i,j} = P[S(t+1) = j | S(t) = i] \tag{1}$$

for $\forall t \in \mathbb{Z}^+$ and $\forall i, j \in \mathbb{Z}_M$. Each state contains a probability density function (pdf) of the multivariate observations. A GMM is a common choice for this purpose [47]. The pdf f_i of an observation $\mathbf{o} \in \mathbb{R}^D$ at state i is

$$f_i(\mathbf{o}) = \sum_{k=1}^{K} w_{i,k} \mathcal{N}(\mathbf{o}; \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k}) \tag{2}$$

where $w_{i,1}, \ldots, w_{i,K}$ are convex coefficients and $\mathcal{N}(\mathbf{o}; \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k})$ denotes a multivariate Gaussian pdf with population mean $\boldsymbol{\mu}_{i,k}$ and covariance matrix $\boldsymbol{\Sigma}_{i,k}$.

978-1-4673-6129-3/13/$31.00 ©2013 IEEE

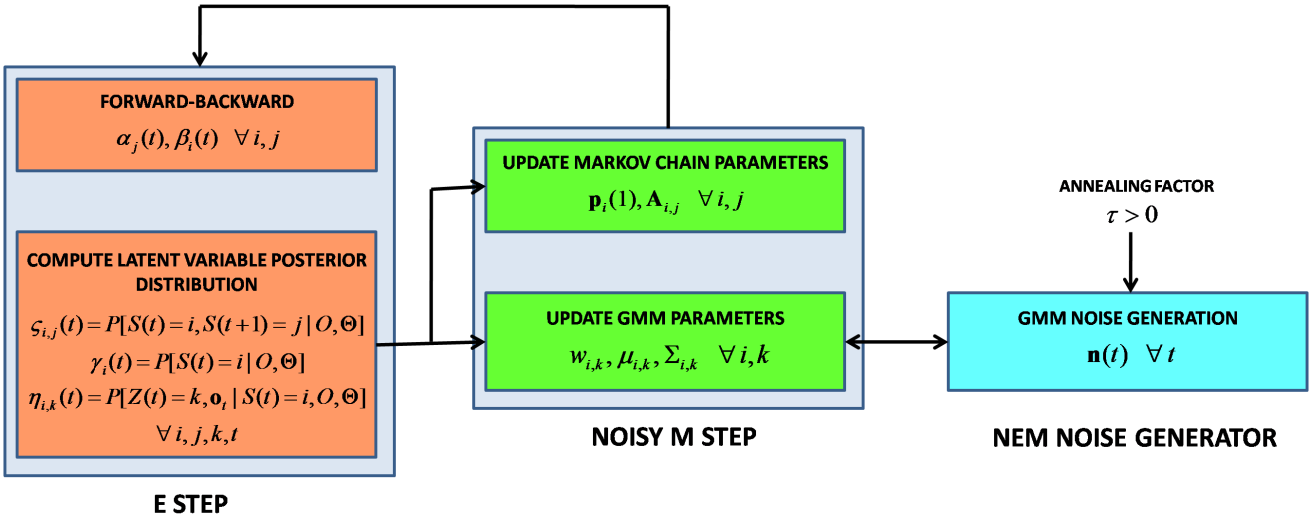


Fig. 1. The training process of the NHMM: The NHMM algorithm adds annealed noise to the observations during the M-step in the EM algorithm if the noise satisfies the NEM positivity condition. This noise changes the GMM covariance estimate in the M-step.

A. The Baum-Welch Algorithm for HMM Parameter Estimation

The Baum-Welch algorithm [3] is an EM approach for maximum likelihood (ML) estimation of HMM parameters. Let $\mathcal{O} = (\mathbf{o}_1, \dots, \mathbf{o}_T)$ denote a multivariate time series of length T. Let $\mathcal{S} = (S(1), \dots, S(T))$ and $\mathcal{Z} = (Z(1), \dots, Z(T))$ be the respective latent state and Gaussian index sequences. Then the ML estimate Θ^* of the HMM parameters is

$$\Theta^* = \arg\max_{\Theta} \log \sum_{\mathcal{S},\mathcal{Z}} P[\mathcal{O}, \mathcal{S}, \mathcal{Z} | \Theta]. \tag{3}$$

The sum over latent variables makes it difficult to directly maximize the objective function (3). EM uses Jensen's inequality [48] and the concavity of the logarithm to obtain the following lower-bound on the observed data log-likelihood $\log P[\mathcal{O}|\Theta]$ at the current parameter estimate $\Theta^{(n)}$:

$$\begin{aligned} \log P[\mathcal{O}|\Theta] &\geq \mathbb{E}_{P[\mathcal{S},\mathcal{Z}|\mathcal{O},\Theta^{(n)}]} \log P[\mathcal{O}, \mathcal{S}, \mathcal{Z}|\Theta] \\ &= Q(\Theta|\Theta^{(n)}) \end{aligned} \tag{4}$$

The complete data log-likelihood for an HMM factors as

$$\begin{aligned} \log P[\mathcal{O}, \mathcal{S}, \mathcal{Z}|\Theta] = &\sum_{i=1}^{M} I(S(1) = i) \log \mathbf{p}_i(1) + \\ &\sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{k=1}^{K} I(S(t) = i, Z(t) = k) \Big\{ \log w_{i,k} + \\ &\log \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k}) \Big\} + \\ &\sum_{t=1}^{T-1} \sum_{i=1}^{M} \sum_{j=1}^{M} I(S(t+1) = j, S(t) = i) \log \mathbf{A}_{i,j} \end{aligned} \tag{5}$$

where $I(.)$ is an indicator function and $\mathbf{p}_i(1) = P[S(1) = i]$. The Q-function requires computing the following sets of variables:

$$\gamma_i^{(n)}(1) = P[S(1) = i | \mathcal{O}, \Theta^{(n)}] \tag{6}$$

$$\eta_{i,k}^{(n)}(t) = P[S(t) = i, Z(t) = k | \mathcal{O}, \Theta^{(n)}] \tag{7}$$

$$\zeta_{i,j}^{(n)}(t) = P[S(t+1) = j, S(t) = i | \mathcal{O}, \Theta^{(n)}] \tag{8}$$

for $\forall t \in \{1, \dots, T\}$, $i, j \in \{1, \dots, M\}$, and $k \in \{1, \dots, K\}$. The Forward-Backward algorithm is a dynamic programming approach that efficiently computes these variables [24]. The resulting Q-function is

$$\begin{aligned} Q(\Theta|\Theta^{(n)}) = &\sum_{i=1}^{M} \gamma_i^{(n)}(1) \log \mathbf{p}_i(1) + \\ &\sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{k=1}^{K} \eta_{i,k}^{(n)}(t) \Big\{ \log w_{i,k} + \log \mathcal{N}(\mathbf{o}_t | \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k}) \Big\} + \\ &\sum_{t=1}^{T-1} \sum_{i=1}^{M} \sum_{j=1}^{M} \zeta_{i,j}^{(n)}(t) \log \mathbf{A}_{i,j}. \end{aligned} \tag{9}$$

Maximizing the auxiliary function $Q(\Theta|\Theta^{(n)})$ with respect to the parameters Θ subject to sum-to-one constraints leads to the re-estimation equations for the M-step at iteration n:

$$\mathbf{p}_i^{(n)}(1) = \gamma_i^{(n)}(1) \tag{10}$$

$$\mathbf{A}_{i,j}^{(n)} = \frac{\sum_{t=1}^{T-1} \zeta_{i,j}^{(n)}(t)}{\sum_{t=1}^{T-1} \gamma_i^{(n)}(t)} \tag{11}$$

$$w_{i,k}^{(n)} = \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t)}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)} \tag{12}$$

$$\boldsymbol{\mu}_{i,k}^{(n)} = \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t) \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)} \tag{13}$$

$$\boldsymbol{\Sigma}_{i,k}^{(n)} = \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t) (\mathbf{o}_t - \boldsymbol{\mu}_{i,k}^{(n)})(\mathbf{o}_t - \boldsymbol{\mu}_{i,k}^{(n)})^T}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)}. \tag{14}$$

We next review the NEM algorithm.

III. The Noisy Expectation-Maximization Theorem

The Noisy Expectation-Maximization (NEM) algorithm [1], [2] modifies the EM scheme and achieves faster convergence times on average. The NEM algorithm injects additive noise into the data at each EM iteration. The noise must decay with the iteration count to guarantee convergence to the optimal parameters of the original data model. The additive noise must also satisfy the NEM condition below. The condition guarantees that the NEM parameter estimates will climb faster up the likelihood surface on average.

A. NEM Theorem

The NEM Theorem [1], [2] states a general sufficient condition when noise speeds up the EM algorithm's convergence to the local optimum of the likelihood surface. The NEM Theorem uses the following notation. The noise random variable $\mathbf{N}$ has pdf $f(\mathbf{n}|\mathbf{o})$. So the noise $\mathbf{N}$ can depend on the observed data $\mathcal{O}$. $\mathcal{L}$ are the latent variables in the model. $\{\Theta^{(n)}\}$ is a sequence of EM estimates for Θ. Θ^* is the converged EM estimate for Θ: $\Theta^* = \lim_{n\to\infty} \Theta^{(n)}$. Define the noisy Q_N function $Q_{\mathbf{N}}(\Theta|\Theta^{(n)}) = \mathbb{E}_{\mathcal{L}|\mathcal{O},\Theta^{(n)}}[\ln f(\mathbf{o}+\mathbf{N},\mathcal{L}|\Theta)]$. Assume that all random variables have finite differential entropy. Assume further that the additive noise keeps the data in the likelihood function's support. Then we can state the NEM theorem [1], [2].

Theorem 1. Noisy Expectation Maximization (NEM) *The EM estimation iteration noise benefit*

$$Q(\Theta_*|\Theta_*) - Q(\Theta^{(n)}|\Theta_*) \quad \geq \quad Q(\Theta_*|\Theta_*) - Q_N(\Theta^{(n)}|\Theta_*) \tag{15}$$

or equivalently

$$Q_N(\Theta^{(n)}|\Theta_*) \quad \geq \quad Q(\Theta^{(n)}|\Theta_*) \tag{16}$$

holds on average if the following positivity *condition holds:*

$$\mathbb{E}_{\mathcal{O},\mathcal{L},\mathbf{N}|\Theta^*}\left[\ln\left(\frac{f(\mathcal{O}+\mathbf{N},\mathcal{L}|\Theta^{(n)})}{f(\mathcal{O},\mathcal{L}|\Theta^{(n)})}\right)\right] \geq 0\,. \tag{17}$$

The NEM Theorem states that each iteration of a suitably noisy EM algorithm gives higher likelihood estimates on average than the noiseless EM algorithm gives at *each* iteration. So the NEM algorithm converges faster than EM does if we can identify the data model. The faster NEM convergence occurs both because the likelihood function has an upper bound and because the NEM algorithm takes larger average steps up the likelihood surface.

Many latent-variable models (such as GMM and HMM) are not identifiable [49], [50] and thus do not have global likelihood optima. The EM and NEM algorithms converge to local optima in these cases. But the added noise in the NEM algorithm may cause the NEM estimates to search nearby local optima. The NEM Theorem still guarantees that NEM estimates have higher likelihood on average than the EM estimates have for such non-identifiable models.

Gaussian mixture model (GMM) parameter estimation greatly simplifies the NEM positivity condition in (17) [1]. Consider the GMM pdf in (2). The model satisfies the positivity condition (17) when the additive noise sample $\mathbf{N} = (N_1, ..., N_D)$ for each observation vector $\mathbf{o} = (o_1, ..., o_D)$ satisfies the following quadratic constraint [1], [2]:

$$N_d\left[N_d - 2\left(\mu_{i,k,d} - o_d\right)\right] \leq 0 \quad \text{for all } k\,. \tag{18}$$

IV. The Noise-Enhanced HMM

The state sequence $\mathcal{S}$ and the Gaussian index $\mathcal{Z}$ are the latent variables $\mathcal{L}$ for an HMM. The noisy Q-function for the NHMM is

$$\begin{aligned} Q_N(\Theta|\Theta^{(n)}) = &\sum_{i=1}^{M}\gamma_i^{(n)}(1)\log\mathbf{p}_i(1) + \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{k=1}^{K}\eta_{i,k}^{(n)}(t) \\ &\left\{\log w_{i,k} + \log\mathcal{N}(\mathbf{o}_t + \mathbf{n}_t|\boldsymbol{\mu}_{i,k},\boldsymbol{\Sigma}_{i,k})\right\} \\ &+ \sum_{t=1}^{T-1}\sum_{i=1}^{M}\sum_{j=1}^{M}\zeta_{i,j}^{(n)}(t)\log\mathbf{A}_{i,j} \end{aligned} \tag{19}$$

where $\mathbf{n}_t \in \mathbf{R}^D$ is the noise vector for the observation $\mathbf{o}_t$. Then the d^{th} element $n_{t,d}$ of this noise vector satisfies the following positivity constraint:

$$n_{t,d}[n_{t,d} - 2(\mu_{i,k,d}^{(n-1)} - o_{t,d})] \leq 0 \quad \text{for all } k \tag{20}$$

where $\boldsymbol{\mu}_{i,k}^{(n-1)}$ is the mean estimate at iteration $n-1$.

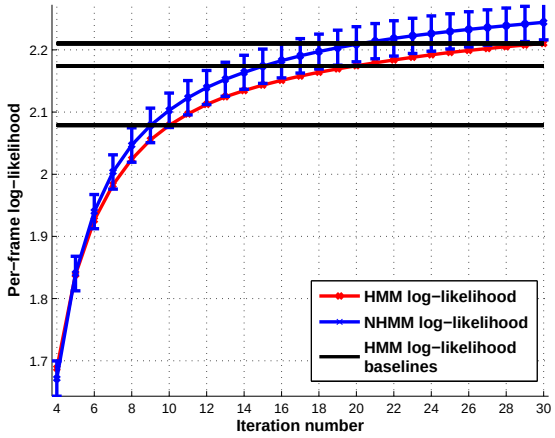
Maximizing the noisy Q-function (19) gives the update equations for the M-step. Only the GMM mean and covariance update equations differ from the noiseless EM because the noise enters the noisy Q-function (19) only through the Gaussian pdf. But the NEM algorithm requires modifying only the covariance update equation (14) because it uses the noiseless mean estimates (13) to check the positivity condition (20). Then the NEM covariance estimate is

$$\boldsymbol{\Sigma}_{i,k}^{(n)} = \frac{\sum_{t=1}^{T}\eta_{i,k}^{(n)}(t)(\mathbf{o}_t + \mathbf{n}_t - \boldsymbol{\mu}_{i,k}^{(n)})(\mathbf{o}_t + \mathbf{n}_t - \boldsymbol{\mu}_{i,k}^{(n)})^T}{\sum_{t=1}^{T}\gamma_i^{(n)}(t)}. \tag{21}$$

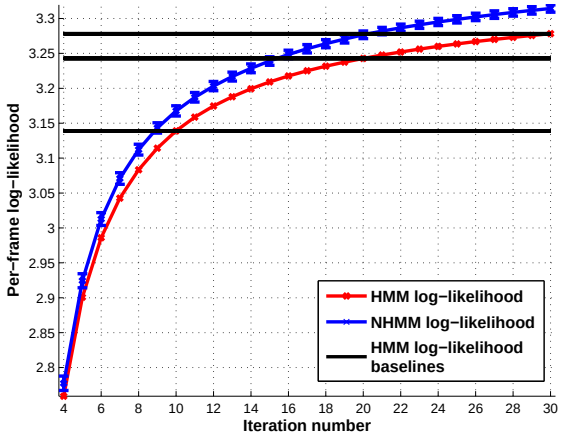
V. Simulation Results

We modified the Hidden Markov Model Toolkit (HTK) [33] to train the NHMM. HTK provides a tool called "HERest" that performs embedded Baum-Welch training for an HMM. This tool first creates a large HMM for each training speech utterance. It concatenates the HMMs for the sub-word units. The Baum-Welch algorithm tunes the parameters of this large HMM.

The NHMM algorithm used (21) to modify covariance matrices in HERest. We sampled from a suitably truncated Gaussian pdf to produce noise that satisfied the NEM positivity condition (20). We used noise variances in $\{0.001, 0.01, 0.1, 1\}$. A deterministic annealing factor $n^{-\tau}$ scaled the noise variance at iteration n. The noise decay rate was $\tau > 0$. We used $\tau \in \{1, \ldots, 10\}$. We then added the noise vector to the observations during the update of the covariance matrices (21).



(a) 16-component GMM at each HMM state



(b) 32-component GMM at each HMM state

Fig. 3. Noise benefit in NHMM training: The plots show the per-frame log-likelihoods for NHMM and HMM with 16 and 32 GMM components per state during successive iterations of Baum-Welch training. The horizontal black lines denote the log-likelihoods for the HMM at iterations 10, 20, and 30. Error bars show one standard deviation above and below the median log-likelihood over 5 NHMM training runs. Noise produces a 1.5% and 1.0% median increase in log-likelihood per iteration over 30 iterations for the NHMM with respective 16 and 32 GMM components.

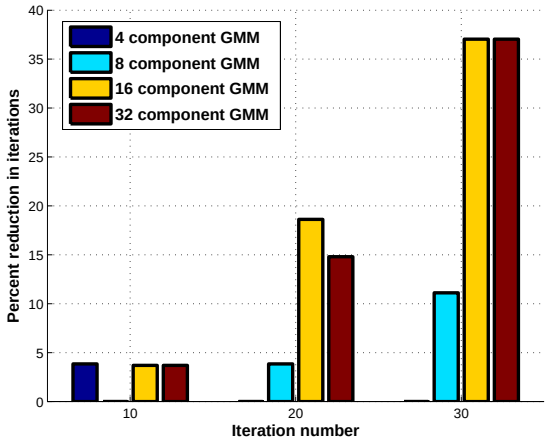


Fig. 2. Reduction in convergence time for the NHMM: The bar graph shows the percent reduction in the number of Baum-Welch iterations with respect to the HMM log-likelihood at iterations 10, 20, and 30. Noise significantly reduces the number of iterations for 8-, 16-, and 32-component GMMs. Noise also produces greater reduction for iterations 20 and 30 due to the compounding effect of the log-likelihood improvement for the NHMM at each iteration. Noise produces only a marginal reduction for the 4-component GMM case at 10 iterations and no improvement for 20 and 30 iterations. This pattern of decreasing noise benefits comports with the data sparsity analysis in [2]. The probability of satisfying the NEM sufficient condition increases with fewer data samples for ML estimation.

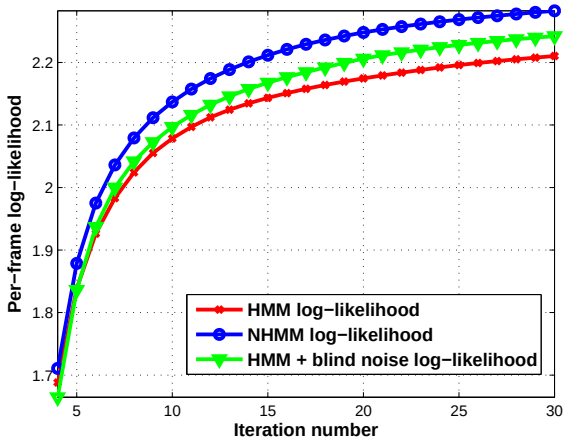


Fig. 4. NHMM versus HMM with "blind noise": This figure compares the per-frame log-likelihoods of an NHMM with a blind noise-added HMM for 16 GMM components per state during successive iterations of Baum-Welch training. We did not constrain the blind noise samples to satisfy the noise benefit inequality in (20). We drew them from an un-truncated Gaussian distribution with identical mean and variance as the NEM noise. The annealed blind noise followed the same cooling schedule as the NEM noise. This figure shows that NHMM gives significantly better log-likelihood than the blind noise HMM.

The simulations used the TIMIT speech dataset [51] with the standard setup in [52]. We parameterized the speech signal with 12 Mel-Frequency Cepstral Coefficients (MFCC) computed over 20-msec Hamming windows with a 10-msec shift. We also appended the first- and second-order finite differences of the MFCC vector with the energies of all three vectors. We used 3-state left-to-right HMMs to model each phoneme with a K-component GMM at each state. We varied K over $\{1, 4, 8, 16, 32\}$ for the experiments and used two performance metrics to compare NHMM with HMM. The first metric was the percent reduction in EM iterations for the NHMM to achieve the same per-frame log-likelihood as does the noiseless HMM at iterations $10, 20$, and 30. The second metric was the median improvement in per-frame log-likelihood over 30 training iterations.

Figure 2 shows the percent reduction in the number of training iterations for the NHMM compared to the HMM log-likelihood at iterations 10, 20, and 30. Noise substantially

Algorithm NHMM Noise-Injection Training

Initialize parameters: $\Theta^{(1)} \leftarrow \Theta_{\text{init}}$

for $n = 1 \rightarrow n_{\max}$ **do**

  function E-STEP($\mathcal{O}, \Theta^{(n)}$)

    for $t = 1 \rightarrow T$, $i, j = 1 \rightarrow M$, and $k = 1 \rightarrow K$ **do**

      $\gamma_i^{(n)}(1) \leftarrow P[S(1) = i|\mathcal{O}, \Theta^{(n)}]$

      $\eta_{i,k}^{(n)}(t) \leftarrow P[S(t) = i, Z(t) = k|\mathcal{O}, \Theta^{(n)}]$

      $\zeta_{i,j}^{(n)}(t) \leftarrow P[S(t+1) = j, S(t) = i|\mathcal{O}, \Theta^{(n)}]$

  function M-STEP($\mathcal{O}, \gamma, \eta, \zeta, \tau$)

    for $i, j = 1 \rightarrow M$ and $k = 1 \rightarrow K$ **do**

      $\mathbf{p}_i^{(n)}(1) \leftarrow \gamma_i^{(n)}(1)$

      $\mathbf{A}_{i,j}^{(n)} \leftarrow \frac{\sum_{t=1}^{T-1} \zeta_{i,j}^{(n)}(t)}{\sum_{t=1}^{T-1} \gamma_i^{(n)}(t)}$

      $w_{i,k}^{(n)} \leftarrow \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t)}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)}$

      $\boldsymbol{\mu}_{i,k}^{(n)} \leftarrow \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t) \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)}$

      $\mathbf{n}_t \leftarrow$ GENERATE-NOISE($\boldsymbol{\mu}_{i,k}^{(n)}, \mathbf{o}_t, n^{-\tau}\sigma_N^2$)

      $\boldsymbol{\Sigma}_{i,k}^{(n)} = \frac{\sum_{t=1}^{T} \eta_{i,k}^{(n)}(t)(\mathbf{o}_t + \mathbf{n}_t - \boldsymbol{\mu}_{i,k}^{(n)})(\mathbf{o}_t + \mathbf{n}_t - \boldsymbol{\mu}_{i,k}^{(n)})^T}{\sum_{t=1}^{T} \gamma_i^{(n)}(t)}$

  function GENERATE-NOISE($\boldsymbol{\mu}_{i,k}^{(n)}, \mathbf{o}_t, \sigma^2$)

    $\mathbf{n}_t \leftarrow \mathcal{N}(0, \sigma^2)$

    for $d = 1 \rightarrow D$ **do**

      if $n_{t,d}[n_{t,d} - 2(\mu_{i,k,d}^{(n-1)} - o_{t,d})] > 0$ for some k **then**

        $n_{t,d} = 0$

    return $\mathbf{n}_t$

reduced the number of iterations for 16- and 32-component GMMs. But it only marginally improved the other cases. This holds because the noise is more likely to satisfy the NEM positivity condition when the number of data samples is small relative to the number of parameters [2]. Figure 3 compares the per-frame log-likelihood of the training data for the HMM and the NHMM. The NHMM has a substantially higher log-likelihood than does the HMM for the 16-and 32-component GMM cases.

Figure 4 shows the comparison between NHMM and HMM with blind noise added to the training data. We did not constrain the blind noise samples to satisfy the noise benefit inequality in (20). The annealed blind noise followed the same cooling schedule and used the same mean and variance as the NEM noise. This figure shows that NHMM gives significantly better log-likelihood than the blind noise HMM. Simulated annealing and blind annealed noise injection also do not guarantee the faster-than-EM convergence that NEM guarantees. The figures in the paper show that NEM gives better likelihoods at each iteration and that NEM converges faster in the long run.

VI. CONCLUSIONS

Careful addition of noise can speed the average convergence of iterative ML estimation for HMMs. The NEM theorem gives a sufficient condition for generating such noise. This condition reduces to a simple quadratic constraint in the case of HMMs with a GMM at each state. Experiments on the TIMIT data set show a significant improvement in per-frame log-likelihood and in time to convergence for the NHMM as compared with the HMM. Future work should develop algorithms to find the optimal noise variance and annealing decay factor. It should also explore noise benefits at other stages of EM training in an HMM.

REFERENCES

[1] O. Osoba, S. Mitaim, and B. Kosko, "Noise Benefits in the Expectation-Maximization Algorithm: NEM theorems and Models," in *The International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2011, pp. 3178–3183.

[2] O. Osoba, S. Mitaim, and B. Kosko, "The Noisy Expectation-Maximization Algorithm," *in review*, 2012.

[3] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, pp. 164–171, 1970.

[4] B. Kosko, *Noise*, Viking, 2006.

[5] A. Patel and B. Kosko, "Levy Noise Benefits in Neural Signal Detection," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 2007, vol. 3, pp. III–1413 –III–1416.

[6] A. Patel and B. Kosko, "Stochastic Resonance in Continuous and Spiking Neurons with Levy Noise," *IEEE Transactions on Neural Networks*, vol. 19, no. 12, pp. 1993–2008, December 2008.

[7] M. Wilde and B. Kosko, "Quantum forbidden-interval theorems for stochastic resonance," *Journal of Physical A: Mathematical Theory*, vol. 42, no. 46, 2009.

[8] A. Patel and B. Kosko, "Error-probability noise benefits in threshold neural signal detection," *Neural Networks*, vol. 22, no. 5, pp. 697–706, 2009.

[9] A. Patel and B. Kosko, "Optimal Mean-Square Noise Benefits in Quantizer-Array Linear Estimation," *IEEE Signal Processing Letters*, vol. 17, no. 12, pp. 1005 –1009, Dec. 2010.

[10] A. Patel and B. Kosko, "Noise Benefits in Quantizer-Array Correlation Detection and Watermark Decoding," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 488 –505, Feb. 2011.

[11] B. Franzke and B. Kosko, "Noise Can Speed Convergence in Markov Chains," *Physical Review E*, vol. 84, no. 4, pp. 041112, 2011.

[12] AR Bulsara, RD Boss, and EW Jacobs, "Noise effects in an electronic model of a single neuron," *Biological cybernetics*, vol. 61, no. 3, pp. 211–222, 1989.

[13] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, "Stochastic resonance," *Reviews of Modern Physics*, vol. 70, no. 1, pp. 223, 1998.

[14] R. Benzi, A. Sutera, and A. Vulpiani, "The mechanism of stochastic resonance," *Journal of Physics A: mathematical and general*, vol. 14, no. 11, pp. L453, 1999.

[15] R. K. Adair, R. D. Astumian, and J. C. Weaver, "Detection of Weak Electric Fields by Sharks, Rays and Skates," *Chaos: Focus Issue on the Constructive Role of Noise in Fluctuation Driven Transport and Stochastic Resonance*, vol. 8, no. 3, pp. 576–587, 1998.

[16] T. R. Albert, A. R. Bulsara, G. Schmera, and M. Inchiosa, "An Evaluation of the Stochastic Resonance Phenomenon as a Potential Tool for Signal Processing," in *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems & Computers*, November 1993, vol. 1, pp. 583–587.

[17] V. S. Anishchenko and T. Kapitaniak, "Chaotic Resonance: Birth of Double-Double Scroll Attractor," in *AIP Conference Proceedings 375: Chaotic, Fractal, and Nonlinear Signal Processing, 1995*, R. A. Katz, Ed., 1996, pp. 420–428.

[18] F. Apostolico, L. Gammaitoni, Marchesoni, and S. Santucci, "Resonant Trapping: A Failure Mechanism in Switch Transitions," *Physical Review E*, vol. 55, no. 1, pp. 36–39, January 1997.

[19] A. S. Asdi and A. H. Tewfik, "Detection of Weak Signals Using Adaptive Stochastic Resonance," in *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, May 1995, vol. 2, pp. 1332–1335.

[20] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, "A Theory of Stochastic Resonance in Climatic Change," *SIAM Journal on Applied Mathematics*, vol. 43, no. 3, pp. 565–578, June 1983.

[21] R. Benzi, A. Sutera, and A. Vulpiani, "The Mechanism of Stochastic Resonance," *Journal of Physics A: Mathematical and General*, vol. 14, pp. L453–L457, 1981.

[22] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *Journal of molecular biology*, vol. 235, no. 5, pp. 1501–1531, 1994.

[23] S.R. Eddy et al., "Multiple alignment using hidden Markov models," in *Proc. ISMB*, 1995, vol. 3, pp. 114–120.

[24] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[25] S.E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer Speech And Language*, vol. 1, no. 1, pp. 29–45, 1986.

[26] J.G. Wilpon, L.R. Rabiner, C.H. Lee, and ER Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[27] S.R. Eddy, "Profile hidden Markov models.," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[28] K. Karplus, C. Barrett, and R. Hughey, "Hidden Markov models for detecting remote protein homologies.," *Bioinformatics*, vol. 14, no. 10, pp. 846–856, 1998.

[29] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. CVPR*. IEEE, 1992, pp. 379–385.

[30] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proc. CVPR*. IEEE, 1997, pp. 994–999.

[31] M.S. Crouse, R.D. Nowak, and R.G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.

[32] R.J. Elliott, L. Aggoun, and J.B. Moore, *Hidden Markov models: Estimation and Control*, vol. 29, Springer, 1994.

[33] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University Engineering Department*, vol. 3, 2002.

[34] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," 2004.

[35] B. Pellom and K. Hacioglu, "Recent improvements in the CU SONIC ASR system for noisy speech: The SPINE task," in *Proc. ICASSP*. IEEE, 2003, vol. 1, pp. I–4.

[36] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney, "The RWTH Aachen University open source speech recognition system," in *Proc. Interspeech*, 2009, pp. 2111–2114.

[37] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlıcek, Y. Qian, P. Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[38] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. SLT*. IEEE, 2010, pp. 97–102.

[39] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz, "BYBLOS: The BBN continuous speech recognition system," in *Proc. ICASSP*. IEEE, 1987, vol. 12, pp. 89–92.

[40] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar, "The AT&T Watson speech recognizer," in *Proc. ICASSP*, 2005, pp. 1033–1036.

[41] J. S. Bridle, "Alpha-Nets: A recurrent neural network architecture with a hidden Markov model interpretation," *Speech Communication*, vol. 9, no. 1, pp. 83–92, 1990.

[42] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-Rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[43] T.N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.R. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*. IEEE, 2011, pp. 30–35.

[44] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[45] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[46] A.R. Mohamed, T.N. Sainath, G. Dahl, B. Ramabhadran, G.E. Hinton, and M.A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5060–5063.

[47] L. Rabiner and B.H. Juang, "Fundamentals of speech recognition," 1993.

[48] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[49] H. Teicher, "On the mixture of distributions," *The Annals of Mathematical Statistics*, pp. 55–73, 1960.

[50] H. Teicher, "Identifiability of finite mixtures," *The Annals of Mathematical Statistics*, vol. 34, no. 4, pp. 1265–1269, 1963.

[51] J.S. Garofolo, *TIMIT: Acoustic-phonetic Continuous Speech Corpus*, Linguistic Data Consortium, 1993.

[52] A.K. Halberstadt and J.R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *Proc. Eurospeech*, 1997, vol. 97, pp. 401–404.