LEARNING SOURCE TRAJECTORIES USING WRAPPED-PHASE HIDDEN MARKOV MODELS

Paris Smaragdis

Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge MA, 02139, USA
paris@merl.com

Petros Boufounos

Research Laboratory of Electronics
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge MA, 02139, USA
petrosb@mit.edu

ABSTRACT

In this paper we examine the problem of identifying trajectories of sound sources as captured from microphone arrays. Instead of employing traditional localization techniques we attack this problem with a statistical modeling approach of phase measurements. As in many signal processing applications that require the use of phase there is the issue of phase-wrapping. Even though there exists a significant amount of work on unwrapping wrapped phase estimates, when it comes to stochastic modeling this can introduce an additional level of undesirable complication. We address this issue by defining an appropriate statistical model to fit wrapped phase data, and employ it as a state model of an HMM in order to recognize sound trajectories. Using both synthetic and real data we highlight the accuracy of this model as opposed to generic HMM modeling.

1. INTRODUCTION

Localization is a problem that has been extensively studied in the audio processing literature [1, 2, 3]. In this paper we will present a modeling approach that leads to a learning methodology, which differs from the traditional time delay or subspace localization methods. Conceptually similar approaches have been presented in the past, however they involved black box training of cross-spectra [4], or straightforward modeling of cross sensor differences [5]. In our work we present a model which is fits sound source trajectories as described from their cross-sensor phase characteristics. We learn and subsequently recognize the physical trajectories of sources as dynamic phase patterns across all frequencies. However because phase is a quantity that is estimated in a wrapped form we had to devise a statistical model to assist the above process, that can take wrapping into account without requiring phase unwrapping. In section 2 we introduce that model and extend it to deal with multivariate time series as an HMM, and in section 3 we show how this model can be used to learn and cluster sound source trajectories and present the relevant results.

2. WRAPPED PHASE MODEL

In this section we define a statistical model for wrapped-phases and wrapped-phase time series. We start from the univariate case in section 2.1 and then extend it for multivariate use and as the state model of an HMM in section 2.2. From here on we will assume that phase wrapping wraps in the interval $[0, 2\pi]$.

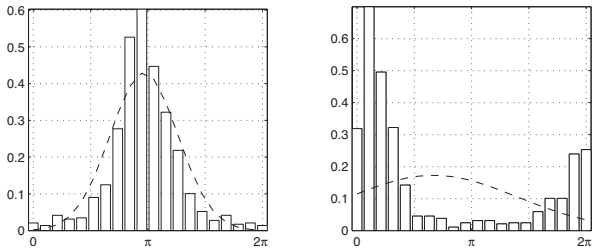

Figure 1: *Histograms of phase data. On the left we have a case that exhibits little phase wrapping which results into data that can be well modeled by a Gaussian distribution. On the right we have a case with severe phase wrapping on which a Gaussian distribution approximation results in a very poor fit for the data.*

2.1. Univariate model

When one makes statistical models the Gaussian distribution is usually a reasonable place to start from. In the case of phase signals though we are faced with an interesting problem due to phase-wrapping. When we model phase with a Gaussian distribution and the mean of the given data is close to 0 or 2π the distribution wraps and becomes bimodal. When this happens a Gaussian model can grossly misrepresent the data. To visualize this consider a histograms of phase data in figure 1. The phase data used for the histograms were the phase differences for specific frequencies between two microphones recording speech. We can see that the histogram on the left is adequately approximated by a Gaussian distribution, however the histogram on the right, exhibiting wrapping, has become bimodal and the fitted Gaussian distribution is a poor description of the data. In order to deal with this issue we will define a proper distribution that explicitly models phase wrapping. To do so we will use the Gaussian distribution as a basis. We will model phase data in its unwrapped form with a Gaussian distribution. We will emulate the wrapping process by replicating and adding the same Gaussian distribution at intervals of 2π:

$$f_x(x) = \begin{cases} \sum_{k=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+k2\pi-\mu)^2}{2\sigma^2}} & \text{if } x \in [0, 2\pi) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

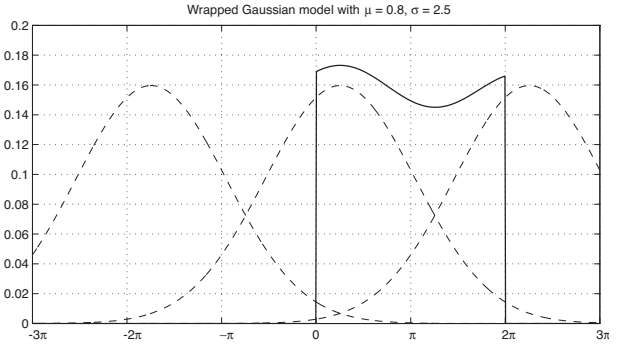

Figure 2: *Illustration of the wrapped Gaussian distribution model. Replicating a Gaussian distribution at intervals of 2π (dashed lines) and summing the result in the interval between $[0, 2\pi)$ (solid line) results into accurate modeling of a wrapped Gaussian distribution.*

The tails of the replicated Gaussian distributions positioned outside the $[0, 2\pi)$ interval will be accounting for the wrapped parts as they enter it. To illustrate consider figure 2 which depicts the distribution of Gaussian distributed phases centered around 0.8. In dotted lines we depict a few of the summed Gaussian distributions used in equation 1. The solid line defined in the interval $[0, 2\pi)$ is their sum and the resulting wrapped distribution. We can see that parts of the central Gaussian distribution that were negative and were wrapped around 2π are being accounted for by the rightmost Gaussian, and the smaller wrapped amount beyond 2π is explained by the left one. The effect of consecutive wrappings of the original data can be respectively explained by Gaussian distributions placed at increasingly distant multiples of 2π. Now that this model is established we move on into describing a procedure to find its optimal parameters to fit a given sample set. To do so we will use Expectation-Maximization (EM) [6]. We will start with a wrapped data set x_i defined in the interval $[0, 2\pi)$, and initial model values μ and σ. The first thing to do is the expectation step, we do so by finding how much each sample belongs into each of the Gaussian distributions in our model:

$$P_{x,k} = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x+k2\pi-\mu)^2}{2\sigma^2}}}{f_x(x)} \tag{2}$$

Using $P_{x,k}$ as a weighting factor we can perform the maximization step and estimate μ and σ. We do so by:

$$\mu = \left\langle \sum_{k=-\infty}^{+\infty} P_{x,k}(x + k2\pi) \right\rangle \tag{3}$$

$$\sigma^2 = \left\langle \sum_{k=-\infty}^{+\infty} P_{x,k}(x + k2\pi - \mu)^2 \right\rangle \tag{4}$$

where $\langle\cdot\rangle$ denotes expectation. Note that the estimate of μ is ambiguous due to wrapping and any solution of the form $\mu + c2\pi, c \in \mathbb{Z}$ is equivalent. For practical implementations summation of an infinite number of Gaussians is obviously an issue. Our experience has been that with $k \in -1, 0, 1$ we can get very good results, and for $k \in -2, -1, 0, 1, 2$ we practically get the same results as with any greater k values. The reason to use large values of k is to account for multiple wraps. However the cases where we get more than three consecutive wraps in our data result from a large data variance with which the data becomes essentially uniform in the defined space of $[0, 2\pi)$. This can be adequately modeled by a large σ and a couple of replicated Gaussians thereby defeating the need of excessive summations over k. In all our experiments in this paper we used $k \in -1, 0, 1$. This truncation of k however introduces a complication in estimating μ. As mentioned above μ is estimated with an arbitrary offset of $c2\pi, c \in \mathbb{Z}$. Now that k is truncated and we have a finite number of Gaussians, it is best to ensure that we have the same number on each side of μ so that we can represent wrappings equally well from both sides. To ensure this we need to make sure that $\mu \in [0, 2\pi)$ which we can easily do by wrapping the estimate we obtain from equation 3.

2.2. Multivariate and HMM extensions

Having the univariate model we can now use it as a basis for a multivariate Hidden Markov Model. First we define the multivariate version. We do so simply by taking the product of the univariate models for each dimension:

$$f_{\mathbf{x}}(\mathbf{x}) = \prod_i f_x(x_i) \tag{5}$$

This essentially corresponds to a diagonal covariance wrapped Gaussian model. A more complete definition is possible by accounting for the full interactions between the variates resulting into the full covariance equivalent, however it is computationally too expensive a model and not required for the purposes of this paper. In this case the parameters that need to be estimated are μ_i and σ_i, for each dimension i. Estimation of the parameters can be done by performing the EM process described above one dimension at a time. Using this as a state model inside an HMM is a straightforward matter. We use the Baum-Welch algorithm [7] to train an HMM that has a wrapped Gaussian as a state model. The only difference from a conventional HMM is that the a posteriori probabilities are computed using the wrapped Gaussian state models and that the state model parameter estimation in the M-step is now defined as:

$$\mu_{i,j} = \left\langle \sum_{k=-\infty}^{+\infty} \gamma_{j,x_i} P_{x_i,k}(x_i + k2\pi) \right\rangle \Bigg/ \sum_{\forall x_i} \gamma_{j,x_i} \tag{6}$$

$$\sigma^2_{i,j} = \left\langle \sum_{k=-\infty}^{+\infty} \gamma_{j,x_i} P_{x_i,k}(x_i + k2\pi - \mu_j)^2 \right\rangle \Bigg/ \sum_{\forall x_i} \gamma_{j,x_i} \tag{7}$$

where γ are the posterior probabilities for each state, j the state index, and i the dimension index. To obtain any reasonable results all of the models we've shown so far are best computed in the logarithmic probability domain to avoid numerical underflows. It is also best if for the first few training iterations all σ^2 are clamped to small values to allow all μ to start converging towards the correct solution. This is because there are strong local optima near 0 and 2π corresponding to a high σ^2. Allowing μ to converge first is a simple way to avoid this problem.

3. LEARNING SOUND TRAJECTORIES

Now that we have a model capable of modeling time series of multidimensional wrapped phase data we can employ it to perform sound trajectory modeling. To do so we will assume that we have

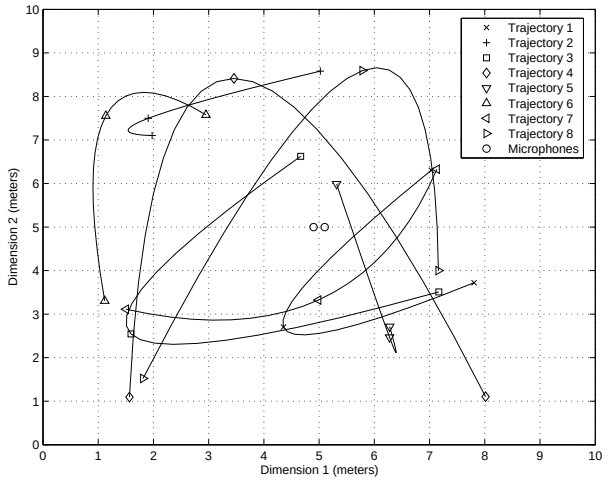

Figure 3: *The eight trajectory types used in the synthetic room examples. The two circles in the middle represent the position of the two microphones in the room.*

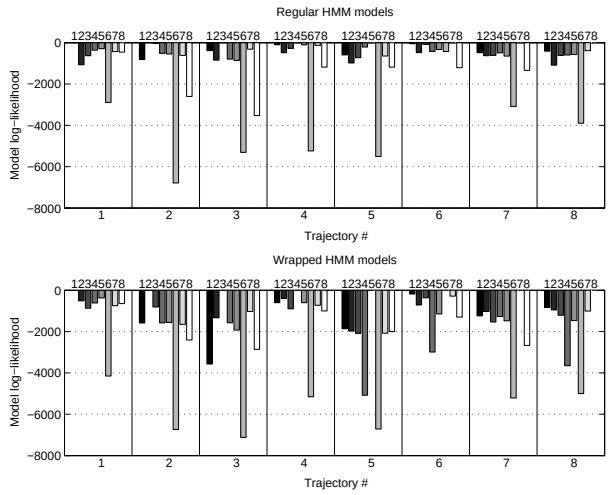

Figure 4: *Model likelihoods for each testing trajectory. Each cluster of bars denotes the likelihood of a trajectory through all the learned trajectory models (trajectory model are denoted on top of bars). The likelihoods are normalized so that the most likely model has zero likelihood (effectively no bar). The top plot shows the results of a regular HMM, whereas the bottom plot shows the results of a wrapped Gaussian HMM.*

a two element microphone array, and that we measure the phase difference in each frequency between the two microphones. To do so we perform a short time Fourier transform on both signals ($F_1(\omega, t)$ and $F_2(\omega, t)$) and compute their relative phase by:

$$\Phi(\omega, t) = \measuredangle \frac{F_1(\omega, t)}{F_2(\omega, t)} \tag{8}$$

Each time instance of Φ was used as a sample point. Subject to symmetry ambiguities, most positions around the two microphones will exhibit a unique phase pattern. Moving sound sources will create time series of such phase patters which we will attempt to model with the framework we just introduced. To avoid measurement noise issues we only used the phase of frequencies ranging from 400Hz to 8kHz. We present results from two experiments, a synthetic one and one with data from a real recording.

3.1. Synthetic results

In this experiment we used the source-image room model [8] to create sound trajectories inside a synthetic room. The room was two-dimensional ($10m \times 10m$) and we used up to 3rd order reflections and a sound absorption coefficient of 0.1. Two cardioid virtual microphones were positioned near the center of the room at positions $(4.9m, 5m)$ and $(5.1m, 5m)$ pointing at opposite directions. In all our examples we used white noise sampled at $44.1kHz$ as the sound source. Eight smooth random trajectories were computed and for each we generated nine similar copies deviating from the originals with a standard deviation of $25cm$. For each trajectory type, we used eight of its copies for fitting a model and then evaluated the likelihood of the ninth one over all these models. The eight types of trajectories are shown in figure 3. We used two training models, a standard Gaussian state HMM and a wrapped Gaussian state HMM as introduced in section 2.2. For both models we trained on eight copies of each of the eight types of trajectories for thirty iterations and used an eight state left-to-right model. Once the models were trained we evaluated the model likelihood of the eight trajectories we have not used yet through all models. The results are shown in figure 4. The groups of bars indicate the likelihoods for each of the test trajectories over all trajectory models. The likelihoods are normalized over the groups so that the more likely model exhibits a likelihood of zero. The wrapped Gaussian HMM models always have the most likely model correspond to the trajectory type, which means that we have assigned all the testing trajectories to the correct type. This is not the case for the regular HMM model which makes classification mistakes due to the inability to model phase accurately. In addition to that the wrapped model provides a statistically more confident classification than the regular model evident by the larger separation of likelihood between the correct and incorrect models.

3.2. Real data results

We repeated the above experiment on data from real recordings. This time we performed stereo recordings in a $3.80m \times 2.90m \times 2.60m$ room. The room featured two glass windows and a whiteboard amounting to about $4.5m^2$ of highly reflective surfaces. Ambient noise in the form of computer fans and air-conditioning amounted to a $-12dB$ noise floor. The recordings were made using a Technics RP-3280E dummy head binaural recording device. We made recordings of eight distinct trajectories, twice using a shaker, producing wide-band noise, and once again using speech. We used the shaker recordings to train our trajectory models and the speech recordings to evaluate their classification accuracy. Just like before we used a sampling rate of $44.1kHz$ and only used the cross-microphone phase measurement of frequencies from $400Hz$ to $8kHz$. The results of this experiment are shown in figure 5. Just like before we can see that the wrapped Gaussian model accurately classifies the speech trajectories to the proper class, whereas the

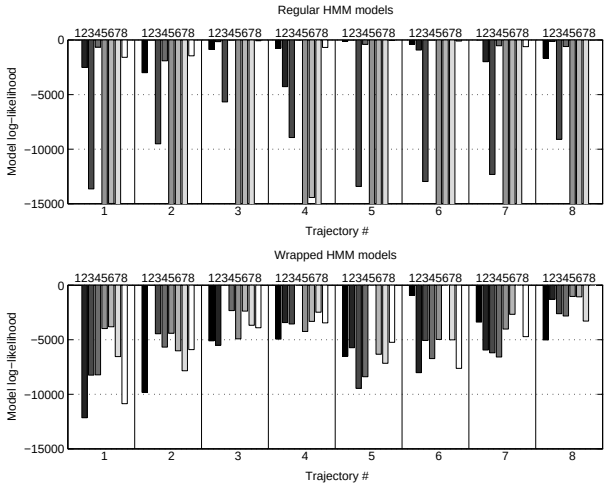

Figure 5: *Model likelihoods for each of the real recording trajectories. The top plot displays the likelihoods using standard HMM models, and the bottom plot using wrapped Gaussian models.*

standard HMM model is hindered by poor data fitting.

3.3. Unsupervised trajectory clustering

So far we used this model for a supervised learning process. We can easily adapt this for clustering applications. Using k-means clusterng and wrapped HMM likelihoods as distances [9], we attempted to cluster the 72 trajectories used in the experiment in section 3.1. We were able to cluster the data in eight clusters with the proper trajectories in each cluster. Using standard Gaussian HMM models for phase we were unable to obtain the correct clustering.

4. CONCLUSIONS AND DISCUSSION

In this paper we presented a statistical model that is able to fit multidimensional wrapped-phase time series. We demonstrated its use in effectively classifying and clustering sound trajectories using microphone arrays. An interesting point that we have observed during our experiments is that since this model is learning phase responses that describe entire environments and not just microphone relationships, we are able to discern locations which traditionally are not discernible using two element arrays. Due to the fact that observed phase measurements are also shaped by the relative positions of all the reflective surfaces and not just the microphones, it is more rare to have ambiguous symmetric configurations that we often see in TDOA based localization. In addition to being able to avoid symmetry ambiguities, this approach is also somewhat resistant to noise. Assuming that the same type of noise is present in the training and the classification examples any phase disruption effects it will have will be learned as part of the model and, assuming they are not dominating, will not detriment classification performance too much. The experiments we presented in this paper make straightforward use of this model, but they are only a starting point as multiple extensions can be realized. Multimicrophone extensions are possible in a variety of ways, most obvious one being defining a model that factors over all microphone pairs. Another simple extension that we have employed takes into account the amplitude difference between two microphones and not just the phase difference. We do so by defining our model in the complex number domain and modeling the real part as a regular Gaussian and the imaginary part as a wrapped Gaussian. We then use this model on the logarithm of the ratio of the spectra of the two signals. The real part of this quantity is the log ratio of the signal energies, and the imaginary part is the cross-phase. That way we model simultaneously both the amplitude and phase differences and with an appropriate microphone setup we are able to discriminate sources in a three dimensional space using only two sensors (similar to how we are able to learn to localize in three dimensions using two ears). Finally we can also perform frequency band selection to make the model more robust. In our examples we used a wide-band training sound which adequately trained all the frequencies, however in cases where the training sounds are not as white then we are better off selecting the frequency bands where both the training and testing sounds have the most energy and evaluating the phase model there. These are just a few of the possible extensions that we have tried, there are many more ways this model can be extended and we hope to address this in future publications.

5. REFERENCES

[1] M.S. Brandstein, J.E. Adcock, and H.F. Silverman, ”A practical time delay estimator for localizing speech sources with a microphone array, *Computer Speech and Language*, vol. 9, pp. 153169, Apr. 1995.

[2] S.T. Birtchfield and D.K. Gillmor, ”Fast bayesian acoustic localization”, *in the proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2002

[3] T. Pham and B.M. Sadler, ”Wideband array processing algorithms for acoustic tracking of ground vehicles”. US Army Research Laboratory, report. Available in: http://www.arl.army.mil/sedd/acoustics/reports.htm

[4] G. Arslan, F.A. Sakarya, and B.L. Evans, ”Speaker Localization for Far-field and Near-field Wideband Sources Using Neural Networks”, *IEEE Workshop on Nonlinear Signal and Image Processing*, 1999.

[5] J. Weng and K. Y. Guentchev, “Three-dimensional sound localization from a compact noncoplanar array of microphones using tree-based learning,” Journal of the Accostic Society of America, vol. 110, no. 1, pp. 310 - 323, July 2001

[6] Dempster, A.P., N.M. Laird, and D.B. Rubin, ”Maximum Likelihood from Incomplete Data via the EM Algorithm,” In *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1-38, 1977

[7] Rabiner, L.R. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 1989.

[8] Allen, J. B. and Berkley, D. A., Image method for efficiently simulating small-room acoustics, JASA Vol. 65, pages 943-950, 1979.

[9] Juang, B.H. and L.R. Rabiner. ”A probabilistic distance measure for hidden Markov models”, *AT&T Technical Journal*, vol. 64 no. 2, February 1985.