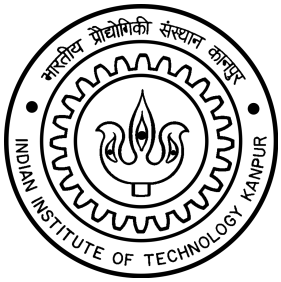
EE602A: Term Project Report



EE602A: Statistical Signal Processing
Instructor: **Prof R. Hegde**, Dept. of EE, IIT-Kanpur
TA: Mr. Ajay Dagar

Submitted by:

Lakshay Garg (13372)
Narsupalli Navin Kumar (13425)
Prakhar Kulshreshtha (13485)
Saurabh Kataria (12807637)

Introduction

Both papers assigned to us have tried to solve the Blind Audio Source Counting problem. The setup is like this: A microphone array records the speech coming from multiple sources (speakers), and the task is to estimate the number of sources from only the speech mixture provided. In both papers, experimental setup is same (described in Simulations section). In [1], STFT is obtained on the microphone signals and DOA for all the coefficients of STFT are obtained. In [2], VEM (Variational Expectation Maximisation) algorithm is run over the cWMM (complex Watson mixture model) and weights of Watson distributions are used to solve the problem.
In [1] and [2], the distributions are modeled using mixture components. Optimal no. of mixture components that best fits the data is reported as the number of sources. Since the distribution is spherical, special MMs are required for modeling. This falls under the realm of circular/directional statistics. In [I], this is done by using Infinite Gaussian Mixture, while in [2], Complex Watson Mixture Models (cWMM) are used for the modeling. More details for each paper, and current implementations are elaborated separately.

I : SOURCE COUNTING IN SPEECH MIXTURES BY NONPARAMETRIC BAYESIAN ESTIMATION OF AN INFINITE GAUSSIAN MIXTURE MODE

Oliver Walter, Lukas Drude and Reinhold Haeb-Umbach

General Ideas of paper:

- Setup: 4x4x3 non-reverberant room
- 3 microphones receive the audio (1 sec length). So we have 3 audio sequences.
- Directions of Arrival (DOA) are found for different frequency components at all the time strides for all the 3 sequences.
- A histogram of these DOAs is plotted.
- Histogram is modeled by a non-parametric Bayesian infinite GMM
- Number of components are reported as number of Sources (speakers).
- A Dirichlet Process prior is employed over mixture components to avoid specifying maximum number of components in advance

- This determines the optimal number of mixture components that best fit the distribution.

Estimating Direction of Arrival(DOA):

At first the STFT for the audio sequences is calculated with a window size W and a a stride s. For each (τ , **f**) we get a coefficients.

$$\mathbf{X}(\tau, f) = \sum_{k=1}^{K} \mathbf{H}_k(f) S_k(\tau, f) + \mathbf{N}(\tau, f), \qquad (1)$$

The coefficients are arranged in a matrix. Using **N** matrices for **R** sensors, we calculate **q** matrix as:

$$q_{ij'}(\tau, f) = \frac{1}{2\pi f_{\text{real}}} \arg\left(X_j(\tau, f) X_{j'}^*(\tau, f)\right)$$

Where q_{ij} is the argument difference for ith and jth sensor.

$$\mathbf{D} = [\ldots, \mathbf{p}_j - \mathbf{p}_{j'}, \ldots]^{\mathrm{T}}$$

Then we calculate **D** matrix as given above, $\mathbf{p}_j$ being the position co-ordinates for jth sensor. $\mathbf{D}^+$ is the Moore Penrose Pseudo Inverse (pinv) of **D**.

$$\begin{bmatrix} \cos\psi(\tau, f) \\ \sin\psi(\tau, f) \end{bmatrix} = v\mathbf{D}^+\mathbf{q}(\tau, f). \qquad (2)$$

Estimate Ψ as tan inverse. Concatenate all Ψs for each time stamp, to obtain the time series vector $\mathbf{d}_n$.

Infinite Gaussian mixture model

The DOA information need to be clustered to provide information about the number of speakers. An infinite Gaussian Mixture model is used to represent the sources. Every Gaussian mixture component represents a source.

While considering the DOA we need to keep in mind that the angles are in the principal range $-\pi$ to π. Hence those DoAa that are close to π wil show bimodal distribution. In other words the data is a wrapped phase. So we consider a wrapped Gaussian mixture model where the probability of an observation with shift k_n to belong to particular l^{th} gaussian component with mean and variance u_l^2 and σ^2 is

$$p\left(d_n|\mu_l, \sigma_l^2, k_n\right) = \frac{1}{\sqrt{2\pi\sigma_l^2}} \exp\left(\frac{-(d_n + 2\pi k_n - \mu_l)^2}{2\sigma_l^2}\right). \quad (3)$$

An indicator variable z_n denotes the weight of a gaussian component towards the nth observation. Hence the entire probability distribution of d_n is

$$p\left(d_n|\boldsymbol{\mu},\boldsymbol{\sigma}^2\right)=\sum_{l=1}^{L}P(z_n=l)\sum_{k_n=-\infty}^{\infty}p\left(d_n|\mu_l,\sigma_l^2,k_n\right). \quad (4)$$

CRP to find z_n

The chinese restaurant process is used to find out the weight denoted by $P(z_n) = \ell$ such that the probability that a old cluster is formed is

$$P(z_{N+1}=l|z_1,\ldots,z_N)=n_l/(N+\gamma). \quad (6)$$

And the probability that a new cluster is chosen is

$$P\ \ p\left(d_n|\boldsymbol{\mu},\boldsymbol{\sigma}^2\right)=\sum_{l=1}^{L}P(z_n=l)\sum_{k_n=-\infty}^{\infty}p\left(d_n|\mu_l,\sigma_l^2,k_n\right). \quad (4)$$

IGMM using CRP process

The first observation is assigned a Gaussian. From the next observation onwards for each observation $P(z_n)$ is found and it is either assigned a new cluster or kept in old cluster.

Parameter Estimation

The parameters u_l^2 and σ^2 are updated through Gibbs sampling and MAP.

In Gibbs sampling, a variable is sampled from the conditional distribution on other variables. To estimate a variable probability distribution of indicator variable is found out

$$P(z_n=l|d_n,\mathbf{d}_{\backslash n},\mathbf{z}_{\backslash n},\mathbf{k}_{\backslash n},\boldsymbol{\Theta}^{(0)})\propto$$
$$P(z_n=l|\mathbf{z}_{\backslash n})p(d_n|\mathbf{d}_{\backslash n},z_n=l,\mathbf{z}_{\backslash n},\mathbf{k}_{\backslash n},\boldsymbol{\Theta}^{(0)}). \quad (8)$$

The parameter Θ_l is integrated out and maximised for k_n from the student's t-distribution

$$P(k_n|d_n,\mathbf{d}_{\backslash n},z_n=l,\mathbf{z}_{\backslash n},\mathbf{k}_{\backslash n},\boldsymbol{\Theta}^{(0)})\propto$$
$$\mathcal{T}(d_n+2\pi k_n;m_l,\xi_l,\eta_l,r_l). \quad (12)$$

and hyper parameters are updated according to the following rule

$$\xi_l=\xi^{(0)}+s_{0,l},$$
$$m_l=(\xi^{(0)}m^{(0)}+s_{1,l})/\xi_l,$$
$$\eta_l=\eta^{(0)}+s_{0,l}/2,$$
$$r_l=r^{(0)}+\left(s_{2,l}+\xi^{(0)}(m^{(0)})^2-\xi_l\mu_l^2\right)/2,$$

Where the sufficient statistics are

$$s_{0,l}=n_l,$$
$$s_{1,l}=\sum_{n:z_n=l}(d_n-2\pi k_n),$$
$$s_{2,l}=\sum_{n:z_n=l}^{N}(d_n-2\pi k_n)^2.$$

Source Counting

After the process, dn observations have gaussians assigned to them each with a particular weight given by indicator variable z_n .We need to remove mixture components with low weights This is done by reducing ɣ by a factor of 100 after a burn in period.

This increases $n_l/(N+\gamma)$ thus increasing weights and reducing $\gamma/(N+\gamma)$, the probability of formation of new cluster

After iterations are over, mixture components with a mean close to the mean of mixture components with higher weights are removed by only keeping those mixture components whose means have the highest probability under their own distribution. MIxture components with σ_l^2 > 10 x (minimum variance of all mixture components) are removed to eliminate noise. Remaining mixture components model the individual DOA's of speaker. Number of mixture components is equal to number of speakers.

II : TOWARDS ONLINE SOURCE COUNTING IN SPEECH MIXTURES APPLYING A VARIATIONAL EM FOR COMPLEX WATSON MIXTURE MODELS

Lukas Drude, Aleksej Chinaev, Dang Hai Tran Vu, Reinhold Haeb-Umbach

General ideas of paper:

1. The paper presents a speaker counting algorithm in a given speech mixture, for both online as well as offline scenarios. The paper is an improvement over [6] in terms of handling noise better.
2. The data is modeled by a mixture of complex Watson distribution (cWMM). And the technique being used to solve for the variables in VEM (variational expectation maximisation)
3. Authors claim through results that their method is better than the DOA based methods.

Coding attempts for this paper:

1. The derivations of the update equations of VEM algorithm is mathematically involved. Also, detailed steps in paper are not provided. It became, as a result, infeasible to code this paper from scratch.
2. A toolbox for cWMM could not be found on Internet. The closest match we discovered were, namely, 'statistics package' for python, 'circular package' for R, and 'circstat package' for MATLAB. They include tools for various circular distributions like Von-Mises, Kent, etc. but Watson (that too complex) distribution or its distribution could not be found.
3. As a result, we understood the concept and results of this paper comprehensively.
4. For a final comment,

Signal model:

$$\mathbf{X}(t,f) = \sum_{k=1}^{K} \mathbf{H}_k(f) S_k(t,f) + \mathbf{N}(t,f), \qquad (1)$$

As can be observed, signal model is same as in paper #1. The observed signal i.e. the one captured at the microphones is a convolution of impulse responses of individual original signal with microphones plus additive noise.
The data is modeled by cWMM as follows:

$$p(\mathcal{Y}|\mathcal{C}) = \prod_{t=1}^{T} \prod_{f=1}^{F} \prod_{k=1}^{K+1} \left(\frac{1}{c_{\mathrm{W}}(\kappa_k)} \mathrm{e}^{\kappa_k |\mathbf{W}_k^{\mathrm{H}} \mathbf{Y}(t,f)|^2} \right)^{c_k(t,f)}. \qquad (3)$$

Total distribution of data is the multiplication of complex Watson distributions in each T-F slots and speech source. K is the maximum number of speakers in the given speech mixture.

Why cWMM is used?
There are a couple of reasons why a complex mixture of Watson distributions is used. They are listed below:

1. cWMM, unlike the GMM maintains all the spatial information in it. So, it is not an approximating technique.
2. Prior information on mode vectors of cWMM can be very helpful for the results.
3. In the expression for cWMM, note that there is a dot product between W and Y. (Y is just a transformation on X). This is helpful because it plays the role of spatial correlation, which maintains the concept of beamforming.

General notes of paper

1. Each point in T-F domain is given a weightage which is modified according to a threshold power P as given below. Note that initially each point is assigned a value of 0.5.

$$A^{(1)}(t,f) = \begin{cases} \frac{1}{2} + \frac{1}{2}a, & \mathbf{X}^{\mathrm{H}}(t,f)\mathbf{X}(t,f) > P, \\ \frac{1}{2} - \frac{1}{2}a, & \mathbf{X}^{\mathrm{H}}(t,f)\mathbf{X}(t,f) < P, \end{cases} \qquad (6)$$

2. W1 vector is initialized by running VEM with size(cWMM)=2, done because EM algo is sensitive to initialisation, after that all W are chosen randomly.

3. Update equation for observation matrix is provided below. This is later used to output a value of number of speakers.

$$A^{(\nu+1)}(t,f) = A^{(\nu)}(t,f)\left(1 - \mathrm{e}^{\kappa_{\mathrm{Re}}\left(\left|\hat{\mathbf{W}}_{\nu}^{\mathrm{H}}\mathbf{Y}(t,f)\right|^2 - 1\right)}\right), \quad (12)$$

4. The online version of the proposed algorithm is simply the frame wise computation of the offline one. However, there are a small number of subtleties involved there.

Results with comments

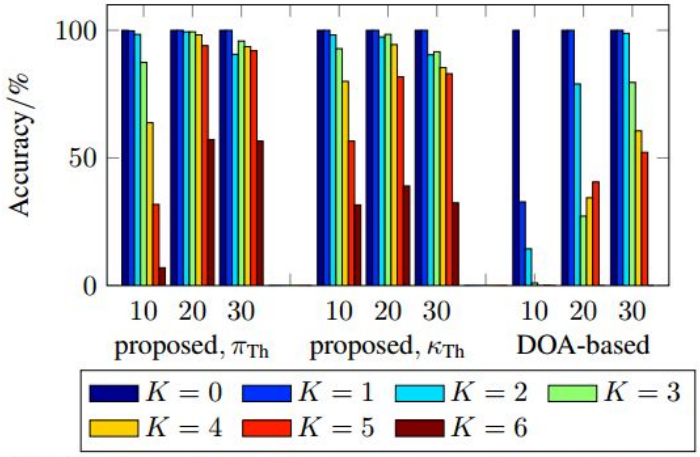


Fig. 2: Comparison of the proposed counting algorithm with the DOA-based algorithm with respect to different thresholds and SNR conditions.

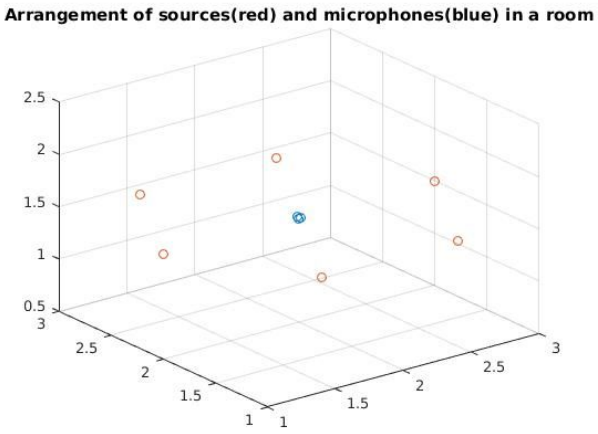
There are three group of results, first two are in this paper, third one is comparison with a DOA based technique. As can be noticed, for a good range of number of speakers, the accuracy for the method with cWMM is outperforming the DOA based one.

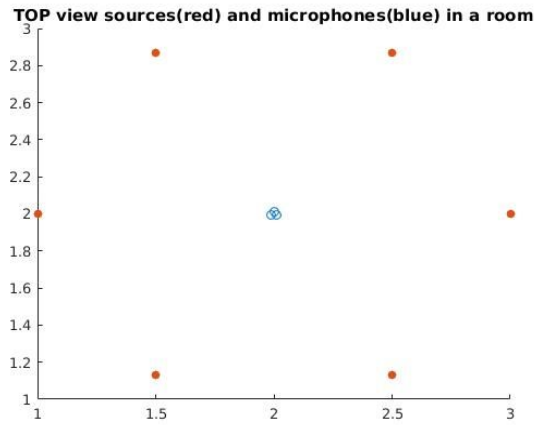
Simulations:

Platform: MATLAB
Toolboxes: MATLAB Signal Processing Toolbox, Kevin Donohues Array Toolbox [4], Dirichlet Process GMM implementation for MATLAB [3]

a) <u>Creating the source and microphone array setup for simulation:</u>

[4] was used to create the setup. Setup of the Simulations are: 4mx4mx3m non reverberant room, WGN (SNR 10-30dB) added, 0 to 6 sources in a radius of 1m (center at 2m,2m), height 1.5m, 3 microphones at same center and height, (max distance between them being 2cm) for receiving source signals.





b) Obtaining the STFT (Short Time Fourier Transform) coefficients for each received signal:

$$\mathbf{X}(\tau, f) = \sum_{k=1}^{K} \mathbf{H}_k(f) S_k(\tau, f) + \mathbf{N}(\tau, f), \qquad (1)$$

An STFT was obtained using a sliding window over the whole signal. This STFT was calculated for all the 3 received signals. The window size is 1024, sampling frequency is 16KHz, and stride is 256. Since DTFT (or STFT) is symmetric, only half the coefficients were used to preserve memory. After that using the approach mentioned in paper, DOA was calculated for each of the (τ , f) coefficients. These DOAs were then concatenated together.

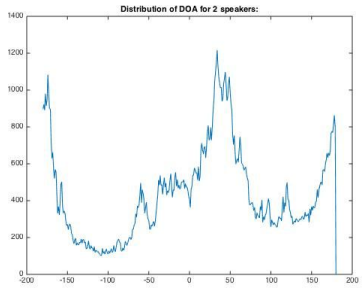
c) Obtaining the DOA for each STFT coefficient:

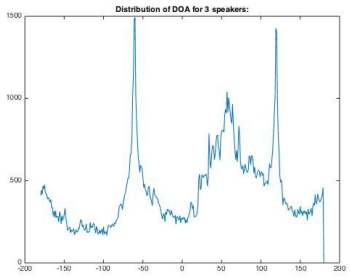
The direction of arrival (DOA) is calculated

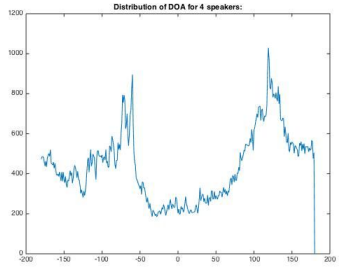
$$q_{ij'}(\tau, f) = \frac{1}{2\pi f_{\text{real}}} \arg\left(X_j(\tau, f) X_{j'}^{*}(\tau, f)\right)$$

d) Histogram of DOA data:

Histograms were obtained for data for different numbers of speakers:







2 Speakers 3 Speakers 4 Speakers

Fig: Distribution of DOA from -180 to 180 for different numbers of speakers

e) Fitting an IGMM on the data:

For IGMM implementation [3] was used. First we tested on a random data of 500 pts for 2 dimensional, two component gaussians:

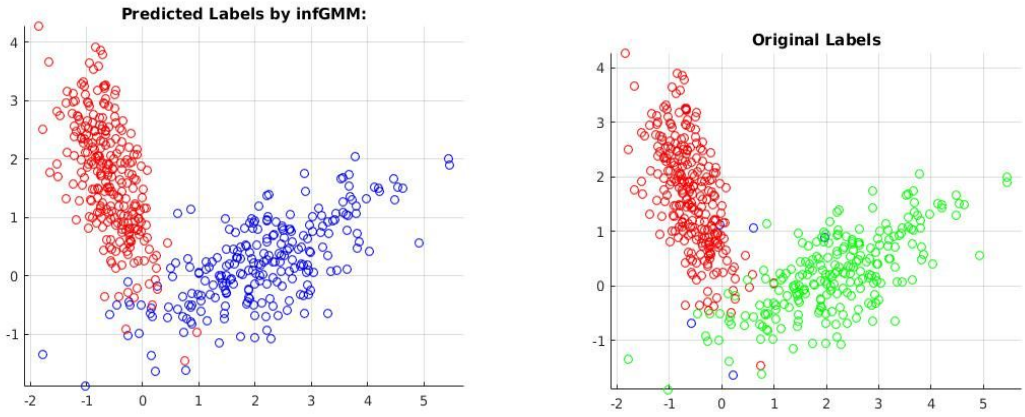
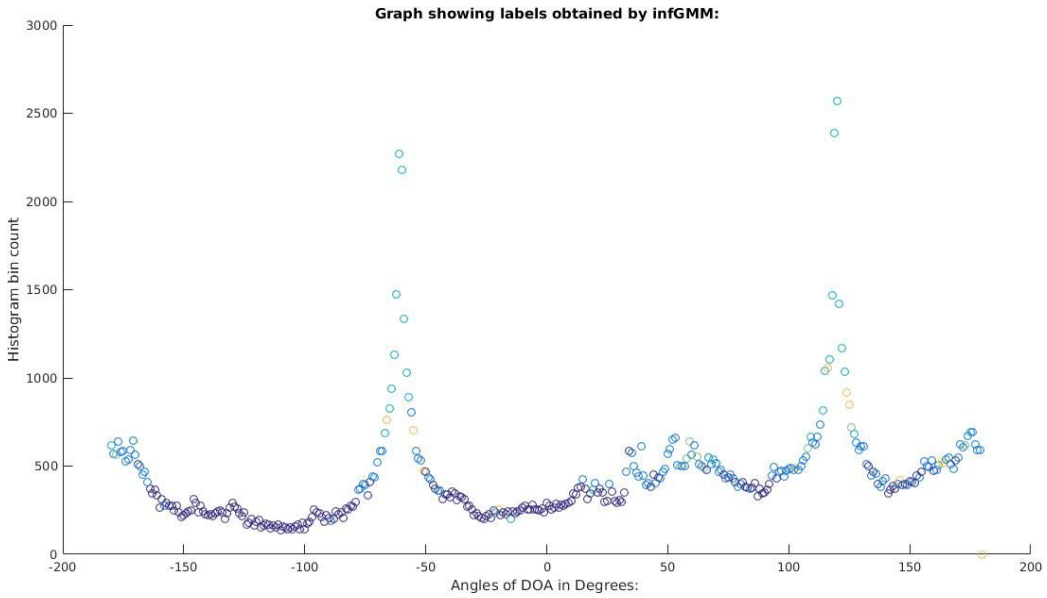


Fig: (Left) shows the original labels and (Right) shows the predicted labels when label-less data was sent to learn IGMM model.

On our data, for 3 speakers arranged in a triangular fashion, the components obtained were:



f) Inferences:

- The number of labels we get on data of 3 speakers is 6. After manual thresholding on weights, we get 3. Authors have also used weight-based thresholding, though they haven't mentioned any automatic heuristic for setting the threshold.
- Hyperparameters of the IGMM have to be tweaked.

- However, the peaks in the histogram are coming out to be 3 i.e. corresponding to 3 speakers.

Possible Improvements:

1. In the first paper, following improvements are possible:
   - Hyperparameter tuning the IGMM on a large dataset such as TIMIT for obtaining better results.
   - Introducing an adaptive threshold learnt on a large dataset, that removes all low weight components from GMM. Currently the authors use a hard-coded threshold for this task.
2. Paper 2:
   - We were unable to find any toolbox for cWMM, even though it is a commonly used Mixture Model. So our proposition would be to create an open source toolbox for the same, that can be published for platforms such as MATLAB, Python, etc.

References:

1. Walter, Oliver, Lukas Drude, and Reinhold Haeb-Umbach. "Source counting in speech mixtures by nonparametric Bayesian estimation of an infinite Gaussian mixture model." *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.
2. Drude, Lukas, et al. "Towards online source counting in speech mixtures applying a variational EM for complex Watson mixture models." *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on*. IEEE, 2014.
3. Dirichlet Process GMM implementation for MATLAB: https://www.mathworks.com/matlabcentral/fileexchange/55865-dirichlet-process-gaussian-mixture-model
4. Kevin Donohues Array Toolbox: www.engr.uky.edu/~**donohue**/audio/**Arrays**/MA**Toolbox**.htm
5. Bijral, Avleen Singh, Markus Breitenbach, and Gregory Z. Grudic. "Mixture of Watson Distributions: A Generative Model for Hyperspherical Embeddings." *AISTATS*. 2007.
6. L. Drude, A. Chinaev, D. H. Tran Vu, and R. Haeb Umbach, Source counting in speech mixtures using a variational em approach for complex watson mixture models, in Proc. ICASSP, May 2014, p. in press