

MODEL BASED MULTIPLE AUDIO SEQUENCE ALIGNMENT

by

Doğaç Başaran

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2002

M.S., Electrical and Electronics Engineering, Boğaziçi University, 2005

Submitted to the Institute for Graduate Studies in  
Science and Engineering in partial fulfillment of  
the requirements for the degree of  
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2015

## MODEL BASED MULTIPLE AUDIO SEQUENCE ALIGNMENT

APPROVED BY:

Prof. Emin Anarım .....

(Thesis Supervisor)

Assoc. Prof. A. Taylan Cemgil .....

(Thesis Co-supervisor)

Prof. Ayşın B. Ertüzün .....

Assoc. Prof. Murat Saraçlar .....

Assoc. Prof. Barış Bozkurt .....

DATE OF APPROVAL: 05.06.2015

## ACKNOWLEDGEMENTS

I would like to express my gratitudes to my thesis supervisor, Prof. Emin Anarım and my co-supervisor, Assoc. Prof. Ali Taylan Cemgil for their invaluable contributions, sharing their wisdom, encouragement and endless support both academically and non-academically. Through this Ph.D. study, whenever I got stuck at a minimum of my own life, they came to my aid to lift me up. I would also like to thank my thesis committee members Prof. Ayşın Baytan Ertüzün, Assoc. Prof. Murat Saraçlar and Assoc. Prof. Barış Bozkurt for their valuable comments that helped me shape this dissertation. I would like to express my gratitude also to Prof. Kadri Özcaldıran for his endless effort to help me in my music career, in my personal life and academic life.

My deepest gratitude goes to my beloved wife Emire and yet-to-born son Emir Deniz. She made me feel so deeply loved, that I can deal with any problem in my life. She is my partner, my otherself, my best friend. I hope that we will share many more years together with such extreme love and with our son that already brought joy to our lives.

I'd also like to deeply thank to my much loving and caring family, my parents Yalçın Başaran and Güzin Başaran and my brother Hasan Başaran. They are always my number one supporters no matter what I do and I feel their love even if we live far away in different cities. I'd like to thank to my aunt Yurdağül Yavuz for her endless love and prayers for me which I believe protect me for a long time. I'd also thank to my aunt Gülsen Abalıoğlu for her support whenever I need.

In the first year of my Ph.D., I also started my professional music career as a bass guitar player in BADEM. Through this hard journey, they witnessed and shared both my darkest and joyful hours. We recorded 4 studio albums, and give over 200 concerts together and in the process, we learned to stand as one. They changed and shaped my life for much better and for that I am truly grateful to Mustafa K. Öztürk, Barış Bahçeci, Mert Özdemir and Emre Yıldız.

I'd like to thank my oldest and dearest friends Kaan Tarıman, Levent Ergün and Onur Uncu for being there for me whenever I need through all my life. Among my best friends, my special thanks goes to my EDKiD family namely Erman Özgüven, Kemal Karaköse, İsmail Erdem Bayıldıran and Doruk Bozdağ. Once a roommate, always a roommate.

I'd like to thank to my friends İpek Şen, Ebru Arısoy, Erinç Dikici, Oya Çeliktutan Dikici, Mehmet Yamaç and İlhan Yıldırım. It is a long journey and sometimes I lost my way but whenever I do, there was always someone that pulled me back in our EE family and without this support, I'd not be able to finish this work. I would also like to thank to Umut Şimşekli, Orhan Sönmez and Barış Kurt for their supportive and valuable friendship. My special thanks goes to İpek Şen due to her warm support through all phases of this thesis. At the time we shared the same laboratory, we shared deep thoughts on life and discuss several subjects through which we try to understand ourselves better. I would also like to thank to Leyla Çeken for her endless support from the beginning till the end of my time in Boğaziçi University.

I would also like to mention some of our good friends and valuable colleagues that we lost along the way through this time namely Hikmet Karayel, İsmail Arı and Kerem Harmanlı. They affected our lives in a way they'd never know and we will always remember them.

This thesis was supported by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610. I'd like to thank for their academic support through most of my Ph.D.

## ABSTRACT

# MODEL BASED MULTIPLE AUDIO SEQUENCE ALIGNMENT

It is increasingly more common that an occasion is recorded by multiple individuals with the proliferation of recording devices such as smart phones. When properly aligned, these recordings may provide several audio and visual perspectives to a scene which leads to several applications in restoring, remastering and remixing frameworks in various fields. In this study, we interpret the problem of aligning multiple unsynchronized audio sequences in a probabilistic framework. In this manner, we propose a novel, model based approach where we define a template generative model. We define 6 different generative models using this template covering basically all kinds of features (real valued, positive, binary and categorical). Proper scoring functions that evaluates the quality of an alignment are derived from each model where we are able to penalize non-overlapping alignments and alignment of a single sequence against a pre-aligned sequences. Having defined a cost or score function, a heuristic sequential search algorithm and a Gibbs sampler approach are proposed to find the optimum alignment of sequences on the surfaces defined by derived score functions. In addition we propose a multi resolution alignment algorithm where we combine Sequential Monte Carlo (SMC) samplers and proposed sequential search method. The models and appropriate features are exhaustively evaluated with artificial and real-life data sets. The simulation results suggest that the approach is able to handle difficult, ambiguous scenarios and partial matchings where simple baseline methods such as correlation fail.

## ÖZET

### MODEL BAZLI ÇOKLU SES DİZİSİ HİZALAMASI

Akıllı telefonlar gibi kayıt yapabilen cihazların artması ile aynı olayın çok sayıda kişi tarafından kayıt edilmesi, günümüzde gittikçe artan bir durum olmaktadır. Uygun hizalandığı takdirde bu kayıtlar, yenileme, yeniden birleştirme ve uyarlama gibi alanlarda çeşitli uygulamalarda kullanılabilir, aynı sahneye çeşitli görsel ve işitsel açılar sağlayacaktır. Bu çalışmada, birbiri ile uyumsuz (hızlı olmayan) çoklu işitsel dizileri hizalama sorunu olasılıksal çerçevede yorumlanmış ve bu bağlamda şablon bir üretimsel model tanımlanarak model tabanlı özgün bir yaklaşım öne sürülmüştür. Bu şablonu kullanarak hemen hemen tüm öznitelik çeşitlerini (gerçek değerli, pozitif, ikili, kategorisel) temelde kapsayan 6 farklı üretimsel model tanımlanmıştır. Bu modellerden hizalamanın kalitesini ölçen uygun puanlama fonksiyonları türetilmiştir. Bu fonksiyonlar dizilerin örtüşmediği durumlarda hizalamaları ve tek bir parçanın önceden hizalanmış parçalara karşı hizalanmasını değerlendirebilmektedirler. Puanlama veya maliyet fonksiyonları tanımlandıktan sonra, bu fonksiyonların oluşturduğu yüzeyde en iyi hizalamanın bulunması için buluşsal ardışık bir arama yöntemi ve Gibbs örnekleme yaklaşımı önerilmiştir. Ek olarak bir çoklu çözünürlük hizalama yöntemi önerilmiş, bu yöntemde ardışık Monte Carlo örnekleme ve önerilen ardışık arama yöntemi birleştirilerek özgün bir yaklaşım geliştirilmiştir. Tüm modeller uygun öznitelikler kullanılarak, yapay ve gerçek veri kümeleri ile değişik senaryolar üzerinde ayrıntılı olarak değerlendirilmiştir. Deney sonuçları bu yaklaşımın ilinti gibi basit, temel metotların yetersiz kaldığı kısmi örtüşmeler, karmaşık ve zor senaryolarda başarılı olduğunu göstermektedir.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
ABSTRACT . . . . .	v
ÖZET . . . . .	vi
LIST OF FIGURES . . . . .	ix
LIST OF TABLES . . . . .	xii
LIST OF SYMBOLS . . . . .	xiii
LIST OF ACRONYMS/ABBREVIATIONS . . . . .	xviii
1. INTRODUCTION . . . . .	1
1.1. Related Work . . . . .	4
1.1.1. Feature Representations for Alignment . . . . .	4
1.1.2. Offset Search for Alignment . . . . .	6
1.2. Contributions of This Dissertation . . . . .	8
2. PROPOSED MODEL . . . . .	11
2.1. Generative Models . . . . .	16
2.1.1. Gamma observation model ( $M_1$ ) . . . . .	16
2.1.2. Gaussian variance observation model ( $M_2$ ) . . . . .	17
2.1.3. Gaussian mean observation model ( $M_3$ ) . . . . .	17
2.1.4. Bernoulli observation model ( $M_4$ ) . . . . .	18
2.1.5. Conditional Bernoulli observation model ( $M_5$ ) . . . . .	18
2.1.6. Multinomial observation model ( $M_6$ ) . . . . .	19
3. PAIRWISE AUDIO SEQUENCE ALIGNMENT . . . . .	21
3.1. Relation between cross correlation and Gaussian mean observation model ( $M_3$ ) . . . . .	24
3.2. Relation between Hamming distance and Conditional Bernoulli obser- vation model ( $M_5$ ) . . . . .	27
3.3. Comparison between correlation and Hamming distance alignment per- formances with $\Phi_{M_3}(\mathbf{r})$ and $\Phi_{M_5}(\mathbf{r})$ scoring function performances under different scenarios . . . . .	30
3.4. Hyperparameter Choices . . . . .	33

3.5. Conditional Bernoulli Observation Model ( $M_5$ ) - Alternative Approach for Pairwise Alignment . . . . .	36
4. MULTIPLE AUDIO SEQUENCE ALIGNMENT . . . . .	39
4.1. Gibbs Sampling with Simulated Tempering . . . . .	39
4.2. Sequential Search Algorithm . . . . .	43
4.3. Multi Resolution Audio Sequence Alignment using Sequential Monte Carlo Samplers . . . . .	46
4.3.1. SMC Samplers for Pairwise Alignment . . . . .	47
4.4. SMC Samplers combined with Sequential Algorithm for Multiple Audio Sequences . . . . .	50
5. EXPERIMENTAL RESULTS . . . . .	54
5.1. Evaluation Criterion . . . . .	55
5.2. Selected Features and Feature Extraction Procedure . . . . .	56
5.3. Hyperparameter Choice . . . . .	58
5.4. Pairwise Audio Sequence Alignment Results . . . . .	61
5.5. Multiple Audio Sequence Alignment Results . . . . .	66
5.5.1. Results for Gibbs Sampler with Simulated Tempering . . . . .	67
5.5.2. Experiment 1: Artificially Produced Data . . . . .	69
5.5.3. Experiment 2: Real-life Data . . . . .	75
6. CONCLUSIONS AND FUTURE RESEARCH . . . . .	80
6.1. Future Research Directions . . . . .	83
APPENDIX A: DERIVATIONS OF SCORE FUNCTIONS $\Phi_{M_i}(\mathbf{r})$ FOR EACH MODEL	
85	
A.1. Gamma Observation Model ( $M_1$ ) . . . . .	85
A.2. Gaussian Variance Observation Model ( $M_2$ ) . . . . .	88
A.3. Gaussian Mean Observation Model ( $M_3$ ) . . . . .	91
A.4. Bernoulli Observation Model . . . . .	96
A.5. Conditional Bernoulli Observation Model ( $M_5$ ) . . . . .	99
A.6. Conditional Bernoulli Observation Model - Alternative Approach for Pairwise Alignment . . . . .	102
A.7. Multinomial Observation Model ( $M_6$ ) . . . . .	107
REFERENCES . . . . .	111

## LIST OF FIGURES

Figure 1.1.	Illustration of multiple audio alignment context with an example. Part <b>A</b> : The unorganised(unaligned) audio sequences $x_1(t)$ , $x_2(t)$ , $x_3(t)$ , $x_4(t)$ and $x_5(t)$ . Part <b>B</b> : The aligned sequences according to each other on the true time line i.e., $x_1(t), x_2(t), x_3(t)$ form cluster 1 and $x_4(t), x_5(t)$ form cluster 2. . . . .	3
Figure 2.1.	Illustration of multiple audio alignment and the model with a toy example. Part <b>A</b> : The unorganised(unaligned) audio sequences; $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$ . Part <b>B</b> : The aligned sequences according to each other on the true time line $\tau$ . Part <b>C</b> : Hidden original source sequence $\lambda$ . . . . .	12
Figure 2.2.	Graphical Model: $x_{k,n,f}$ is the observed coefficient of sequence $k$ , $\lambda_{\tau,f}$ is hidden coefficient and $r_k$ is the alignment of sequence $k$ , round rectangles are plates that replicates the nodes inside i.e., $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}$ or $x_{1,1,1}, x_{1,1,2}$ etc. . . . .	14
Figure 3.1.	Illustration of pairwise alignment with a toy example. $x_1$ and $x_2$ are two sequences to be aligned (inside the rectangle). $\tau$ is the global time index and $r$ is the current alignment of $x_2$ . $\Phi(r)$ is shown for all relative shifts. . . . .	23
Figure 3.2.	Subparts of $\Phi_{M_3}(r)$ for two overlapping sequences, from top to bottom $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$ , $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$ , correlation term and resulting $\Phi_{M_3}(r)$ . . . . .	26
Figure 3.3.	Normalized Hamming distance for all relative shifts $r_2$ between two overlapping sequences. . . . .	28

Figure 3.4.	The contribution of aligned equal bits and aligned different bits versus $w$ parameter on computation of $\Phi_{M_5}(r)$ score function. . . .	29
Figure 3.5.	Pairwise alignments using $R(r)$ , $H(r)$ , $\Phi_{M_3}(r)$ and $\Phi_{M_5}(r)$ for three different scenarios. The left column (a,d,g,j): Two overlapping sequences (high SNR), the middle column (b,e,h,k): Two non-overlapping sequences (high SNR), the right column (c,f,i,l): Two overlapping sequences (Low SNR). . . . .	32
Figure 3.6.	3D plot of $\Phi_{M_5}(r_2, w)$ over $r_2$ and $w$ . . . . .	34
Figure 3.7.	$\Phi_{M_5}(r)$ results for two non-overlapping sequences computed for two different $w$ parameters (a) $\Phi_{M_5}(r)$ , $w = 0.75$ (b) $\Phi_{M_5}(r)$ , $w = 0.6$ (c) Zoomed version of $\Phi_{M_5}(r)$ , $w = 0.6$ . . . . .	35
Figure 4.1.	Gibb's Sampling with simulated tempering algorithm for multiple audio sequence alignment model. . . . .	41
Figure 4.2.	Sequential Alignment Method. . . . .	45
Figure 4.3.	(a) Illustration of model for pairwise case with a toy example. Two observed sequences $\mathbf{x}_1, \mathbf{x}_2$ , hidden sequence $\lambda$ (b) Modified toy example for the modified model for half resolution in pairwise case. .	48
Figure 4.4.	The story of a particle. . . . .	50
Figure 4.5.	Smoothed Bridge Distribution through each stage from top view. .	51
Figure 5.1.	Extraction of feature $F_4$ - Real Features. . . . .	57
Figure 5.2.	Extraction of feature $F_5$ - Binary Features. . . . .	57

Figure 5.3.	Sources and features illustration: (a) Three aligned time domain sequences (b) Extracted positive spectral difference ( $F_2$ ) for aligned sequences (c) Extracted MFCC coefficients ( $F_3$ ) for aligned sequences.	59
Figure 5.4.	3D plot of $\Phi_{M_5}(r_2, w)$ over $r_2$ and $w$ .	60
Figure 5.5.	Alignment on synthetic data. (a) Generated Sequences $\mathbf{x}_1$ and $\mathbf{x}_2$ (b) Gibbs sampler (SA) alignment estimation with 2500 epochs(c) $r_2$ estimate at each epoch.	68
Figure 5.6.	Alignment on real data using Gibbs: 1000 Epochs, 20 times. (a) Aligned time domain signals $x_1(t), x_2(t)$ and $x_3(t)$ (b) Aligned features $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ .	69
Figure 5.7.	The Scoring Function $\Phi_{M_1}(\mathbf{r})$ for all possible values of alignments.	70
Figure 5.8.	Performances of different model-feature pairs in mean alignment scores and standard deviations for rock, jazz, classical music and speech a) for high SNR cases ( $\approx 10\text{dB}$ ) b) for low SNR cases ( $\approx -3\text{dB}$ ).	71
Figure 5.9.	Alignment performances of sequential method with $M_5F_5$ , SMC Sampler with $\Phi_{M_5}(\mathbf{r})$ and baseline method in mean alignment scores and standard deviations for rock, jazz, classical music and speech a) for high SNR cases ( $\approx 10\text{dB}$ ) b) for low SNR cases ( $\approx -3\text{dB}$ ).	74
Figure 5.10.	The number of exact, 1-bit difference and 2-bit difference hash matches between two noisy versions of the same song with varying SNR levels.	75

## LIST OF TABLES

Table 2.1.	Prior and likelihood distributions for each model. . . . .	16
Table 3.1.	Similarity of two binary sequences $x_{1,0:3,1:2}$ and $x_{2,0:3,1:2}$ aligned at $r_1 = 2$ and $r_2 = 3$ . . . . .	28
Table 5.1.	Hyperparameter choices for each model for low SNR and high SNR cases. . . . .	61
Table 5.2.	Alignment performances of each model-feature pair for 20 real data sets. . . . .	63
Table 5.3.	Alignment performances using cross correlation and Hamming distance among 20 real data sets. . . . .	64
Table 5.4.	False Positive/Negative Analysis. <i>O</i> : Overlap at the true alignment, <i>NO</i> : Not overlapping, <i>FP</i> : False positive, <i>FN</i> : False negative. . .	65
Table 5.5.	p-values for comparison of $M_5F_5$ performance with all other model-feature pair performances in low SNR case. . . . .	73
Table 5.6.	p-values for comparison of $M_5F_5$ performance with all other model-feature pair performances in high SNR case. . . . .	73
Table 5.7.	Number of clips taken by each camera for data set 1. . . . .	76
Table 5.8.	Alignment performances ( $\Omega(\hat{r}_{1:K})$ ) of model-feature pair $M_5F_5$ , SMC Sampler with $\Phi_{M_5}(\mathbf{r})$ and baseline correlation method for data set 1 and data set 2. . . . .	78

## LIST OF SYMBOLS

$A(\mathbf{x}, \beta_\lambda, \alpha, r)$	Part of $\Phi_{M_3}(r)$ that does not include correlation
$A0(\mathbf{x}, \beta_\lambda, \alpha)$	Subpart of $A(\mathbf{x}, \beta_\lambda, \alpha, r)$ that is independent of alignment $r$
$A1(\mathbf{x}, \beta_\lambda, \alpha, r)$	Subpart of $A(\mathbf{x}, \beta_\lambda, \alpha, r)$
$A2(\mathbf{x}, \beta_\lambda, \alpha, r)$	Subpart of $A(\mathbf{x}, \beta_\lambda, \alpha, r)$
$B_i(\cdot)$	Backward transition kernel at $i$ 'th step in SMC samplers
$\mathcal{B}$	Beta Distribution
$\mathcal{BE}$	Bernoulli Distribution
$C_i$	$i$ 'th cluster of aligned sequences
Dir	Dirichlet Distribution
$f$	Frequency bin index
$F$	Number of frequency bins
$F_1$	Feature type 1
$F_2$	Feature type 2
$F_3$	Feature type 3
$F_4$	Feature type 4
$F_5$	Feature type 5
$F_6$	Feature type 6
$F_7$	Feature type 7
$F_8$	Feature type 8
$\mathcal{G}$	Gamma Distribution
$H(r)$	1-normalized Hamming distance
$H_0$	Null hypothesis
$\mathcal{IG}$	Inverse Gamma Distribution
$k$	Index of sequence i.e., $k$ 'th sequence
$K$	Number of unaligned sequences
$K'$	Number of aligned sequences in a group
$\mathcal{K}_i(\cdot)$	Forward transition kernel at $i$ 'th step in SMC samplers
$L$	Resolution level in SMC samplers
$\mathcal{M}$	Multinomial Distribution

$m_k^1$	Volume variable for sequences
$m_k^2$	Volume variable for noise components
$\mathcal{N}$	Normal Distribution
$n$	Local time index
$N_k$	Length of $k$ 'th observed sequence
$n_L$	Local time index at resolution level $L$
$\mathcal{P}$	Conditional Bernoulli Distribution
$p(\lambda)$	Prior distribution of hidden feature sequence $\lambda$
$p(\lambda, \Theta)$	Prior distribution of hidden feature sequence $\lambda$ with hyperparameters $\Theta$
$p(\lambda_{1:T})$	Joint prior distribution of all hidden feature coefficients $\lambda_{1:T}$
$p(r_k)$	Prior distribution of alignment of $k$ 'th sequence
$p(\mathbf{r} \mathbf{x})$	Posterior distribution of alignments $\mathbf{r}$
$p(\lambda, \mathbf{r} \mathbf{x})$	Posterior distribution of alignments $\mathbf{r}$ and hidden sequence $\lambda$
$p(\mathbf{r} \mathbf{x}, \Theta)$	Posterior distribution of alignments $\mathbf{r}$ conditioned both on observations and hyperparameters
$p(r_k \cdot)$	Full conditional density of $r_k$
$p(\lambda_\tau \cdot)$	Full conditional density of $\lambda_\tau$
$x$	Observed feature sequence
$p(\mathbf{x}, \mathbf{r})$	Joint distribution of alignments and observations
$p(\mathbf{x} \mathbf{r})$	Likelihood distribution of alignments
$p(\mathbf{x} \mathbf{r}, \Theta)$	Likelihood distribution of alignments conditioned also on the hyperparameters
$p(\mathbf{x} \lambda, \mathbf{r}, \Theta)$	Observation model
$Q$	Smoothing kernel
$r$	Alignment of a sequence
$r_1$	Alignment of sequence 1
$r_2$	Alignment of sequence 2
$\hat{r}_i$	Estimated alignment for $i$ 'th sequence
$\mathbf{r}$	All alignments i.e., $r_{1:K}$
$\mathbf{r}^*$	Optimum alignments
$r_{1:K}$	All alignments

$r_{1:K}^*$	Optimum alignment of all sequences
$r_k$	Alignment random variable of $k$ 'th sequence
$r_k^*$	Alignment estimate of $k$ 'th sequence
$r_s^{(i)}$	Sample alignment at $i$ 'th step with sample index $s$
$P$	Unaligned sequence list in sequential alignment algorithm
$R(r)$	Cross correlation
$\mathcal{R}_\infty$	Camera 1
$\mathcal{R}_\infty$	Camera 2
$\mathcal{M}_\parallel$	Number of sequences in the cluster $k$
$R_{x_1, x_2}$	Cross correlation between sequences $x_1$ and $x_2$
$S_{\mathbf{x}_i, \mathbf{x}_j}$	Similarity functions between sequences $\mathbf{x}_i$ and $\mathbf{x}_j$
$T$	Length of hidden feature sequence $\lambda$
$T_0$	Computation time of $\Phi(r)$ for all alignment estimates at the highest resolution
$T_L$	Computation time of $\Phi(r)$ for all alignment estimates at resolution level $L$
$U$	Unclustered sequence list in sequential alignment algorithm
$\mathcal{U}(r, \tau)$	An indicator function that shows if a coefficient exists at time $\tau$ according to the alignment $r$
$\mathbf{w}$	Hyperparameter of the conditional Bernoulli observation model $M_5$
$w_{0,0}$	Probability that $\lambda_{\tau,f} = 0$ and the observed coefficient $x_{k,n,f} = 0$ given that $x_{k,n,f}$ is aligned at time $\tau$
$w_{1,0}$	Probability that $\lambda_{\tau,f} = 1$ and the observed coefficient $x_{k,n,f} = 0$ given that $x_{k,n,f}$ is aligned at time $\tau$
$w_{0,1}$	Probability that $\lambda_{\tau,f} = 0$ and the observed coefficient $x_{k,n,f} = 1$ given that $x_{k,n,f}$ is aligned at time $\tau$
$w_{1,1}$	Probability that $\lambda_{\tau,f} = 1$ and the observed coefficient $x_{k,n,f} = 1$ given that $x_{k,n,f}$ is aligned at time $\tau$
$w_{i,j}$	Probability that $\lambda_{\tau,f} = i$ and the observed coefficient $x_{k,n,f} = j$ given that $x_{k,n,f}$ is aligned at time $\tau$
$w$	Probability that $\lambda_{\tau,f} = x_{k,n,f}$ given that $x_{k,n,f}$ is aligned at time $\tau$

$x_k(t)$	Observed sequence k in time domain
$\mathbf{x}$	$\mathbf{x}_{1:K} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$ Matrix of feature sequences
$\mathbf{x}_k$	Feature sequence k: $x_{k,0:N_k-1,1:F}$
$x_{k,n}$	$n$ 'th coefficient of $k$ 'th sequence
$x_{k,n,f}$	Coefficient of $k$ 'th sequence at time $n$ and frequency bin $f$
$\alpha$	Hyperparameter of observation models in $M_1$ and $M_3$
$\alpha_\lambda$	Hyperparameter of prior distributions in $M_1, M_2, M_4$ and $M_5$
$\beta$	Parameter for simulated tempering
$\beta_\lambda$	Hyperparameter of prior distributions in $M_1, M_2, M_3$ and $M_4$
$\gamma_i$	Intermediate distribution at $i$ 'th step
$\epsilon$	Threshold for length of overlap
$\phi(\hat{r}_i, \hat{r}_j)$	Indicator function that determines if sequences $i$ and $j$ are mutually correctly aligned.
$\Phi(\mathbf{r}; \Theta)$	Score function which is defined as logarithm of the likelihood of alignments i.e., $\log p(\mathbf{x} \mathbf{r}, \Theta)$
$\Phi(r_{1:K})$	Score function of all alignments
$\Phi_L$	Score function at resolution level $L$
$\Phi(\mathbf{r})$	Score function of all alignments
$\Phi(r)$	Score function of only one alignment $r$
$\Phi_{M_1}(\mathbf{r})$	Score function of alignments derived from $M_1$
$\Phi_{M_2}(\mathbf{r})$	Score function of alignments derived from $M_2$
$\Phi_{M_3}(\mathbf{r})$	Score function of alignments derived from $M_3$
$\Phi_{M_4}(\mathbf{r})$	Score function of alignments derived from $M_4$
$\Phi_{M_5}(\mathbf{r})$	Score function of alignments derived from $M_5$
$\Phi_{M_5}(r, w)$	Score function of alignments and hyperparameter $w$ derived from $M_5$
$\Phi_{M_{5.1}}(\mathbf{r})$	Score function of alignments derived from $M_5$ - alternative approach
$\Phi_{M_6}(\mathbf{r})$	Score function of alignments derived from $M_6$
$\Phi_{M_i}(\mathbf{r})$	Score function of alignments derived from $M_i$
$\lambda$	The hidden feature sequence

$\lambda_\tau$	The hidden feature coefficient at time $\tau$
$\lambda_{\tau,f}$	The hidden feature coefficient at time $\tau$ and frequency bin $f$
$\lambda_{1:T}$	All hidden feature coefficients
$\pi_{k,\tau}$	The probability that sequence $k$ is aligned at time $\tau$
$\Omega$	Alignment performance criterion
$\tau$	The global time index
$\theta$	A random variable
$\Theta$	Hyperparameters

**LIST OF ACRONYMS/ABBREVIATIONS**

1-D	One Dimensional
2-D	Two Dimensional
ABS	Absolute Value
CQT	Constant Q Transform
dB	Decibell
DCT	Discrete Cosine Transform
DNA	Deoxyribonucleic Acid
DFT	Discrete Fourier Transform
<i>ESS</i>	Effective Sample Size
<i>FN</i>	False Negative
<i>FP</i>	False Positive
LPCC	Linear Predictive Coding Coefficients
MAA	Multiple Audio Alignment
MCMC	Markov Chain Monte Carlo
MCS	Multiple Camera Synchronization
MFCC	Mel-Frequency Cepstral Coefficients
MP	Matching Pursuit
ms	Mili Second
<i>NO</i>	Not Overlapping
<i>O</i>	Overlapping
SMC	Sequential Monte Carlo
SNR	Signal To Noise Ratio
ST	Simulated Tempering
STFT	Short Time Fourier Transform
U.S.	United States
XOR	Exclusive OR

## 1. INTRODUCTION

With the proliferation of recording devices, an increasing number of people regularly capture audio and video in special occasions like weddings, parties and holiday trips. As a result, a single event can be simultaneously recorded by multiple individuals (such as using mobile phones) creating multiple views, wide coverage and listening perspectives to a scene. This data is typically made accessible mostly through media sharing sites (such as youtube) however in unorganised form. This problem, which is referred to as Multiple Audio Alignment (MAA), is defined in this context as follows: There are  $K$  audio clips with no information available on the offset settings of these sequences. Then the problem is to find these offsets of multiple audio clips relative to each other on the generic time line.

Our motivation in dealing with multiple audio alignment problem is that properly aligning unorganized recordings would lead to several use cases in source separation, restoration, remixing or remastering frameworks. Such a scenario might occur for example in a concert hall during a performance. Assume that some of the audience record their favourite parts of the concert with recording devices of varying quality. These audio clips, each of which are recorded from a different perspective, would also have different amplitude levels and noise. A possible application might be collecting these unsynchronized audio recordings on a website and try to produce a full recording of the performance by precisely aligning these sources on the generic time line.

*Multiple-Camera Synchronization*(MCS) or *Multiple-Video Synchronization* is one of the main application areas where multiple audio alignment is excessively used. MCS in professional setting is performed by a genlock signal or a clapper board providing a reference for all the cameras. However in a non-professional environment such reference mechanisms do not exist. In those cases, the unorganized video signals are synchronized using their corresponding audio parts utilizing the fact that audio and video are already synchronized with each other. Such an application can be found in [1] where the video clips that are taken by audience in the same concert are synchronised from their audio

to obtain a full-clip of one song. In [2], over 700 YouTube videos related to a U.S. presidential inauguration are used to restore the U.S. President’s speech. An application of automatic video remixing can be found in [3], where the system automatically creates remixes from videos recorded by mobile devices. There are also commercially available products for video synchronization such as *AudioAlign*<sup>1</sup>, *PluralEyes* and *DualEyes* from *SingularSoftware*<sup>2</sup>.

Applications of video synchronization can also be found in forensics field. Analysis on properly aligned multiple video recordings of civilians and surveillance cameras that are recorded during a criminal event, might provide extra leads for law enforcement. Such an application is mentioned in [4]. Another potential application is audio forensic enhancement where multiple recordings are time aligned to isolate the desired voice from background activity [5].

With a wide area of applications, alignment of multiple audio sequences is a very challenging task due to the following reasons:

- A clean, original source track against which individual sequences can be matched is absent but only noisy observations are available.
- The recording devices can have different characteristics and recordings can have dramatically different sound quality
- None of the sequences have to cover the entire timeline
- The audio sequences may or may not overlap

To illustrate a possible scenario in alignment framework, an example is given in Figure 1.1. In this example, there are five unaligned (unorganized) sequences that are labeled as  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ ,  $x_4(t)$  and  $x_5(t)$ . When the sequences are aligned on the global time line, the first three sequences ( $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ ) overlap with each other and they form a cluster (Cluster 1). The last two sequences ( $x_4(t)$ ,  $x_5(t)$ ) overlap with each other but not with other sequences so they form another cluster (Cluster 2).

---

<sup>1</sup><https://www.audioalign.com/>

<sup>2</sup>In this thesis, we work in collaboration with Singular Software company. Available at, <http://www.singularsoftware.com>

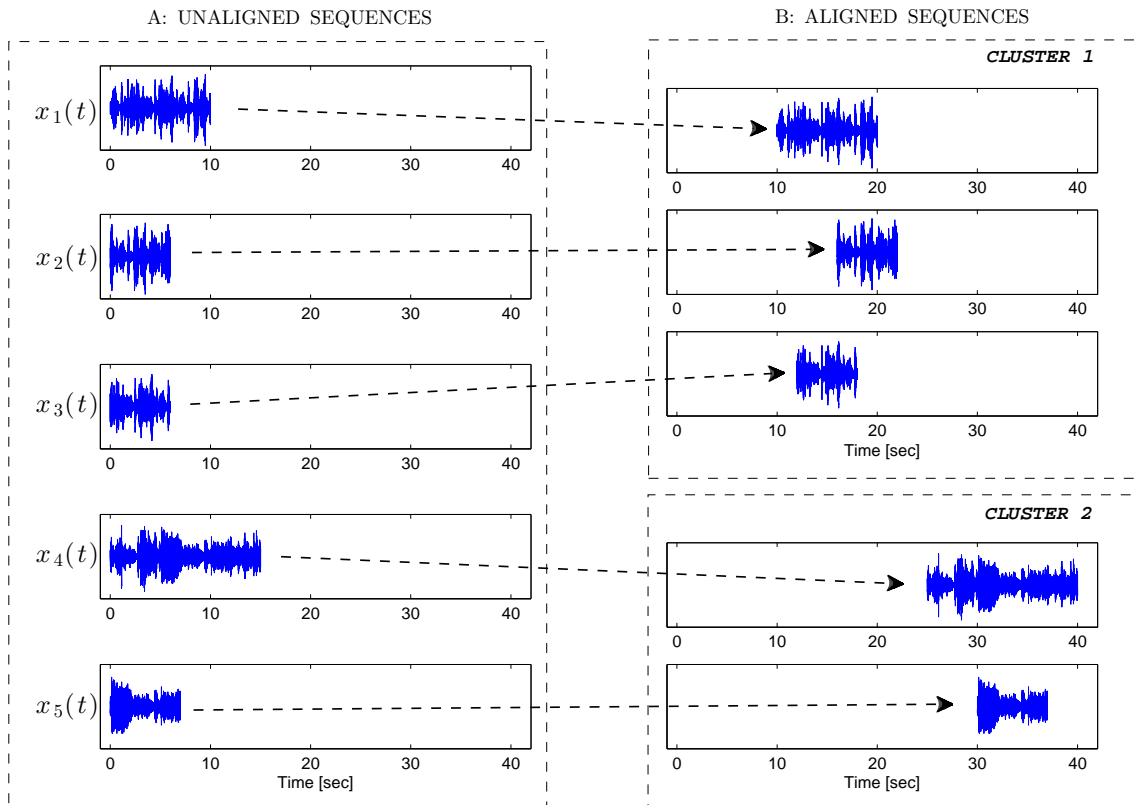


Figure 1.1. Illustration of multiple audio alignment context with an example. Part **A**: The unorganised(unaligned) audio sequences  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ ,  $x_4(t)$  and  $x_5(t)$ . Part **B**: The aligned sequences according to each other on the true time line i.e.,

$$x_1(t), x_2(t), x_3(t) \text{ form cluster 1 and } x_4(t), x_5(t) \text{ form cluster 2.}$$

Whilst interesting, we are not focusing on other audio alignment scenarios in this thesis such as aligning audio recordings of different interpretations (covers) of the same piece of music [6], [7], aligning musical recordings to symbolic representations [8], [9] and aligning midi files to audio recordings [10].

Audio alignment is closely related to audio matching or fingerprinting framework. In principle, similar algorithms are employed for both problems, however they differ in their goals. In fingerprinting, the aim is to match an unidentified, possibly noisy audio signal to a large clean audio database. It is widely used in music identification services such as *Shazam* [11] where the metadata of an audio signal (e.g. song title, artist name) is retrieved from a short segment of it. Fingerprinting is also used to

monitor radio broadcasts for copyright purposes [12–14]. There exists highly robust audio fingerprinting methodologies with high matching performance under very noisy conditions [11–24]. A review of audio fingerprinting methods can be found in [25].

This approach to multiple sequence alignment is reminiscent to genetics, where long DNA strands are assembled from shorter sequences, a process that is called shotgun sequencing [26]. Image stitching is another visual analogy in which multiple images of the same scene taken from slightly different perspectives are assembled into a full panoramic view [27].

The next section will focus on the extensive review of the existing literature related to audio alignment framework.

## 1.1. Related Work

In literature, audio alignment problem is tackled in a twofold manner: First the audio signals are represented with robust features against various types of noise and then search algorithms are employed over these representations to find the best offset setting of sequences usually utilizing similarity or cost functions.

### 1.1.1. Feature Representations for Alignment

A suitable representation of the audio for alignment needs to be easy to compute, require low memory and be robust against degradations. Transform domain representations are better suited for such requirements. Thus, the alignment is usually applied on time-frequency representations such as short-time Fourier Transform (STFT) rather than on raw audio data.

Audio fingerprints were originally proposed for audio matching purposes. However, due to their compact structure and robustness against noise, they are also widely used in multiple audio alignment. A comparison between matching the raw audio signals and matching audio fingerprints is presented in [28]. The audio fingerprints are

presumably used for synchronisation or alignment purposes rather than matching for the first time. Note that a fingerprint acts like a sketch or a feature sequence, thus we use the terms feature sequence and fingerprint as synonyms in this work.

One of the most cited audio representations is proposed by Wang [11]. In this method, the local frequency peaks are identified in the STFT of a given audio segment under the assumption that they are robust in the presence of noise. The "landmarks" that are constructed by the time and frequency differences between the current peak and a few peaks in an adjacent target zone are then used as fingerprints. In [1] and [29], the procedure in [11] is followed to represent the audio parts of video clips which are then used for synchronization of the video. In [2], matching pursuit (MP) algorithm is employed for feature extraction where the algorithm iteratively selects MP basis functions called atoms corresponding to the most energetic points in the signal. Then following [11], the landmarks are formed from the center frequencies and the time difference between atoms. Recently in [21], a new audio fingerprint named "MASK: Masked Audio Spectral Keypoints" is proposed which also uses the spectrogram peaks combined with a spectral masking procedure.

Another popular audio representation was presented by Haitsma in [15]. This method extracts the signs of energy differences between pairs of band energies both in time and frequency axes of the spectrogram to obtain 32-bit binary features which are called sub-fingerprints. Then 256 sub-fingerprint form a fingerprint. This methodology is used in [30] and [31] that is a follow up work of [28] for synchronization of multiple video clips.

Recently, computer vision techniques are applied on the 2-D time-frequency representations of 1-D audio signals in extraction of fingerprints by viewing 2-D representations as images. In [22], a learning process is introduced into feature selection by selecting representative filters which are actually Viola and Jones features for face detection [24]. A similar vision based approach named "waveprint" is presented in [23], where Haar wavelet coefficients are used for time-frequency representations.

Beside mentioned features, there are also mel-frequency cepstral coefficients (MFCC) [16], Linear predictive coding coefficients (LPCC) [32], positive spectral difference [6], [33], chroma-based features [19], [20] and constant Q transform (CQT) [17]. In this work, we use various types of features i.e., positive, real valued, binary, categorical. The features can be categorized and listed as follows:

- Positive feature sets
  - (i)  $F_1$ : Energy in sub-bands
  - (ii)  $F_2$ : Positive Spectral Difference [6], [33]
- Real valued feature sets
  - (i)  $F_3$ : MFCC [16]
  - (ii)  $F_4$ : First difference of the energy in subbands both through frequency and time
  - (iii)  $F_7$ : First difference of the energy in subbands of the spectrum through time
- Binary feature sets
  - (i)  $F_5$ : Thresholding the first difference of the energy in subbands both through frequency and time [15]
  - (ii)  $F_6$ : The thresholded energy in sub-bands
- Categorical feature set
  - (i)  $F_8$ : Quantized coefficient energy in each subband

### 1.1.2. Offset Search for Alignment

In the multiple audio alignment setup, the aim is to find the offset of each audio file (relative distances on the time line according to each other). There are several offset search methods that utilize well-known fingerprinting methods for the task.

In [28], the fingerprinting scheme in [15] is applied to find the offset in the following way: Fingerprint blocks (consist of 256 sub-fingerprint of length 11.6ms) are generated from two audio clips, then the Hamming distance is computed for all possible lags between sequences and the offset is chosen as the point where the minimum distance occurs. A pre-classification is applied on the fingerprint blocks for audio classes

(silence, music, speech, noise and crowd) to fasten the procedure by comparing only the fingerprints of the same classes. The method is further improved in [30] and [31] by constructing a *Look Up Table* or a *Hash Table* that relates audio clips and corresponding fingerprints. Each fingerprint acts as a dictionary entry and exact matches of sub-fingerprints of different sequences are searched assuming that there will be at least one exact sub-fingerprint match between two matching audio sequences which is debatable<sup>3</sup>.

A fingerprinting approach similar to [30] have been proposed in [1]. In this method, the time-stamped fingerprints are treated as simple hash values. To find the offset, the number of matching hash values of two given audio clips is computed and if it exceeds a pre-defined threshold, then the audio clips are assumed to be matching. The offset between matched time-stamps is then used as the offset between audio clips. Cross-correlation is applied for a finer alignment. In [2], sparse landmarks are extracted from audio clips using Matching Pursuit and a hash table is constructed where the landmark - audio clip relations are held similar to [30] and [1]. The number of exactly matching landmarks are used to decide the offset in same way as [1].

A method with several improvements to [28] is proposed in [29] where generalized cross-correlation is used to find the offset between two audio sequences. The generalized cross-correlation is applied on the sparse landmarks that are extracted following [11] from each audio clip. One major contribution of this approach is that the matching sequences are merged together using an ad hoc method and next sequence is correlated with the merged matching sequences. By this way, the computational time is minimized on the search for offsets.

As mentioned fingerprinting methods (hashing strategies) are widely used for audio alignment purposes due to their robust and low memory requiring structure. However, there are several obstacles in dealing with multiple alignment using fingerprinting methods:

---

<sup>3</sup>An analysis on number of exact fingerprint matches between two matching sequences is given in Section 5.5.2

- All the fingerprinting methods rely on finding exact fingerprint matches between matching audio clips. But due to the variability of sound quality and noise in each recording, an exact match in the fingerprints may not occur.
- A threshold has to be determined for the matching/non-matching decision on the number of exact hash matches which depends highly on the data.
- There is a fixed but rather small dictionary for hash values therefore many spurious matches are likely to occur.
- Most of the methods (except [29]) are designed to match fingerprints of two audio clips. By this way, in multiple audio alignment setup where there are  $K$  audio clips, each pair of clips have to searched separately.

As a result, the methods employed for audio fingerprinting are only indirectly applicable to multiple audio alignment.

Recently, some multimodal (combination of audio and video based methods) synchronization methods have been proposed in [30], [34] and [35]. In these methods, some visual strategy i.e., flash lights in the video, is used to match signals then the alignment is obtained by applying simple similarity measures on the sequences such as correlation and Hamming distance.

In practice, the alignment problem can be tackled using any deterministic similarity based approach such as in [28], [29] and [32] which have proven to be robust under noisy conditions. However, there are limitations to the success of such methods. First of all, important threshold parameters need to be selected to choose overlapping and non-overlapping sequences; however such choices are very difficult to make optimally as they depend highly on data (similar to hashing strategies). Secondly, for multiple sequence alignment i.e.,  $K$  number of sequences, aligning each pair requires in the order of  $O(K^2)$  computation which can be prohibitive even for small size fingerprints.

## 1.2. Contributions of This Dissertation

Major contributions of this thesis can be stated as follows:

- A novel probabilistic interpretation and a model based approach for the multiple audio sequence alignment problem is proposed. Unlike previous studies in the literature, a wide range of audio features are applicable because the type of audio feature is determined by the choice of the observation model.
- We define a template generative model which is quite generic and can be used with different choices of distributions. To this end, we define six generative models using the template, to cover basically all types of the features including real, positive, non-negative, binary and categorical feature sets.
- Proper scoring functions are derived from each model. In literature, simple similarity measures such as cross correlation and Hamming distance are widely used for alignment purposes. Our score functions have two main advantages over those methods:
  - (i) Score functions are able to quantify automatically (without using an ad hoc threshold) a matching / non-matching (or overlap / no overlap) of sequences whereas cross-correlation and Hamming distance are not.
  - (ii) Score functions facilitate the alignment of a single sequence against a group of pre-aligned sequences whereas correlation and Hamming distance can only be used to align two sequences.
- Instead of hashing and exact match strategies that are widely used in literature, we propose a sequential search algorithm. In hashing methods, alignment for all pairs in the data set have to be computed and merged in an ad hoc way. However, we provide a method where we solve the problem without finding all pair alignments.
- A multi resolution alignment procedure is proposed on top of the sequential search algorithm where we utilise a *coarse to fine* structure. To our knowledge, standard search algorithms in literature work on single resolution level.

The outline of the rest of the thesis is as follows: In Chapter 2, we introduce the model based approach to the multiple alignment problem and describe a method to measure the quality of an alignment. We define six generative models to cover a wide range of feature types. The model is analyzed for pairwise alignments and the relationship between several models and similarity measures i.e., correlation and Hamming distance, are revealed in Chapter 3. We describe our proposed offset search

methods for multiple alignment in Chapter 4. In this chapter, we first provide a Gibbs sampling approach then describe the sequential search algorithm. The multi resolution alignment procedure using sequential Monte Carlo samplers is explained in the end of this chapter. In Chapter 5, we give an evaluation criterion, feature and hyper parameter choices and then the models are exhaustively evaluated using both artificial and real-life data sets. A comparison of baseline and proposed methods is also given in Chapter 5. We conclude the thesis and suggest future directions in Chapter 6. The derivations of score functions for each model is given in Appendix A.

## 2. PROPOSED MODEL

In this chapter, we introduce our probabilistic approach to the multiple audio sequence alignment problem: a 'template' generative model is defined and the associated graphical model is given. The central theme of the proposed model is as follows: *Given the correct alignment  $r$ , observed feature sequences (fingerprints)  $x$  are noisy realizations from a common but unobserved parameter sequence  $\lambda$ .* Formally, we will define a probability model called the observation model denoted as  $p(\mathbf{x}|\mathbf{r}, \lambda, \theta)$  where  $\theta$  are the hyperparameters. We will choose a suitable prior over the unobserved parameter sequence denoted by  $p(\lambda)$ . Using this model we can compute the quality of an alignment  $\mathbf{r}$  by the posterior expression,

$$p(\mathbf{r}|\mathbf{x}, \theta) \propto p(\mathbf{x}|\mathbf{r}, \theta) = \int d\lambda p(\mathbf{x}|\lambda, \mathbf{r}, \theta)p(\lambda, \theta) \quad (2.1)$$

where  $p(\mathbf{r}|\mathbf{x}, \theta)$  is the posterior and  $p(\mathbf{x}|\mathbf{r}, \theta)$  is the likelihood of alignments  $\mathbf{r}$ . We refer to the logarithm of the right hand side of the expression in Equation 2.1 as the scoring function  $\Phi(\mathbf{r}; \theta) = \log p(\mathbf{x}|\mathbf{r}, \theta)$ . Later, hyperparameters  $\theta$  are assumed to be fixed hence we have dropped the dependence on  $\theta$  and use  $\Phi(\mathbf{r})$  instead.

Below, we will describe the model that is illustrated by a toy example given in Figure 2.1 in more detail. In Figure 2.1, we show a hidden unobserved parameter sequence  $\lambda$  (Part C), and four feature sequences  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  and  $\mathbf{x}_4$  (Part A and Part B). In part A, the sequences are shown in an unorganised or unaligned form, on the other hand in part B, the sequences are aligned according to each other on the global time line. Here, each square block in a sequence represents a coefficient of a sequence and the color of each square encodes its value (as in a heat map). We denote the hidden parameter vector with  $\lambda_{1:T}$ . Here  $\tau = 1 \dots T$  is a global time frame index. There are four clips observed in Figure 2.1 and the feature vector of the  $k$ 'th clip is denoted as  $\mathbf{x}_k$ . The length of the feature vector of the  $k$ 'th clip is denoted as  $N_k$ . In this example,  $T = 19$ ,  $N_1 = 5$ ,  $N_2 = 7$ ,  $N_3 = 6$  and  $N_4 = 5$ . Here,  $n$  is a local time frame index for each sequence and the coefficient of the  $k$ 'th clip at local time  $n$  is denoted by  $x_{k,n}$

(depicted as a square). The alignment variable (starting point) for the  $k$ 'th clip is denoted as  $r_k$ . For example,  $\mathbf{x}_2$  is aligned at global time  $\tau = 5$  therefore  $r_2 = 5$ . In this toy scenario, none of the sequences cover the entire time line. The  $\mathbf{x}_1, \mathbf{x}_2$  and  $\mathbf{x}_3$  clips overlap with each other at several points i.e.,  $x_{1,4}, x_{2,1}$  and  $x_{3,3}$  coincide at global time  $\tau = 6$ . It can be observed that each of these coefficient values are close to each other since they are observations of a common source  $\lambda_6$ . In reality, the actual parameter sequence  $\lambda$  is also unknown, hence we will integrate over this quantity.

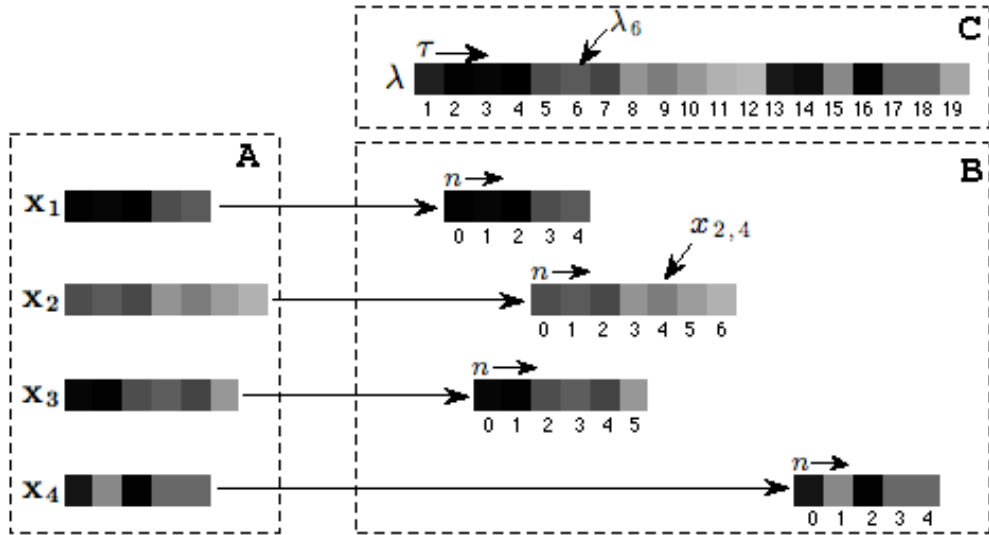


Figure 2.1. Illustration of multiple audio alignment and the model with a toy example. Part **A**: The unorganised(unaligned) audio sequences;  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  and  $\mathbf{x}_4$ . Part **B**: The aligned sequences according to each other on the true time line  $\tau$ . Part **C**: Hidden original source sequence  $\lambda$ .

Making the model more precise, the template generative model is defined as follows:

$$\begin{aligned}
\lambda_{1:T} &\sim p(\lambda_{1:T}) \\
r_k &\sim p(r_k) = \prod_{\tau=1}^{T-N_k+1} \pi_{k,\tau}^{[r_k=\tau]} \\
x_{k,n} &\sim p(x_{k,n}|r_k, \lambda_{1:T}) = \prod_{\tau=1}^T p(x_{k,n}|r_k, \lambda_\tau)^{[n=\tau-r_k]}
\end{aligned} \tag{2.2}$$

where  $[\cdot]$  is the indicator function which is equal to one if the expression inside is true. The probability of the  $k$ 'th clip to start at time  $\tau$  is represented with  $\pi_{k,\tau}$ . In general, we do not have any information on sequence alignments, so it would be fair to assume that each  $r_k$  is independent and uniformly distributed. Here  $p(\lambda_\tau)$  is the prior distribution of a single parameter,  $p(r_k)$  is the prior distribution alignment of  $k$ 'th sequence and  $p(x_{k,n}|r_k, \lambda_\tau)$  is the observation model of the  $n$ 'th feature of the  $k$ 'th sequence. Here, the  $[n = \tau - r_k]$  expression in the observation model indicates that if  $x_{k,n}$  is aligned to time  $\tau$ , then it only depends on the hidden coefficient  $\lambda_\tau$ , hence each observation coefficient is conditioned on a different hidden coefficient. Note that the hidden parameters  $\lambda_{1:T}$  are assumed to be a-priori independent so the prior distribution is decomposed as follows:

$$p(\lambda_{1:T}) = \prod_{\tau=1}^T p(\lambda_\tau) \tag{2.3}$$

In audio, features for a single sequence are usually not 1-dimensional hence a single scalar  $\lambda_\tau$  is not enough for describing a single time slice. So we can denote the hidden parameter at the time instant  $\tau$  and for the frequency band  $f$  as  $\lambda_{\tau,f}$ . Related graphical model is shown in Figure 2.2. Note that the round rectangles in Figure 2.2 are plates that replicates the nodes inside [36]. For the sake of simplicity, we denote  $\mathbf{r} = r_{1:K}$ ,  $\mathbf{x} = x_{1:K,0:N_k-1}$  and the  $f$  index is omitted in the rest of the thesis document.

In the alignment problem, the aim is to estimate the most likely alignment of the clips that we denote as  $r_{1:K}^*$ . This is the prime mode of the joint posterior probabil-

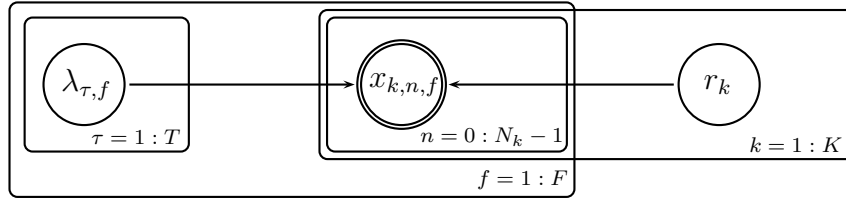


Figure 2.2. Graphical Model:  $x_{k,n,f}$  is the observed coefficient of sequence  $k$ ,  $\lambda_{\tau,f}$  is hidden coefficient and  $r_k$  is the alignment of sequence  $k$ , round rectangles are plates that replicates the nodes inside i.e.,  $\lambda_{1,1}, \lambda_{1,2}, \lambda_{2,1}$  or  $x_{1,1,1}, x_{1,1,2}$  etc.

ity  $p(\mathbf{r}|\mathbf{x})$ . Assuming no prior information on alignments, likelihood  $p(\mathbf{x}|\mathbf{r})$ , posterior  $p(\mathbf{r}|\mathbf{x})$  and joint distribution  $p(\mathbf{x}, \mathbf{r})$  are all proportional as shown in Equation 2.4.

$$p(\mathbf{x}|\mathbf{r}) \propto p(\mathbf{r}|\mathbf{x}) \propto p(\mathbf{x}, \mathbf{r}) \quad (2.4)$$

Hence, one can use the likelihood distribution  $p(\mathbf{x}|\mathbf{r})$  to estimate  $\mathbf{r}^*$  that can be derived by integrating the joint conditional distribution  $p(\mathbf{x}, \lambda_{1:T}|\mathbf{r})$  over latent variables  $\lambda_{1:T}$  as follows:

$$p(\mathbf{x}|\mathbf{r}) = \int d\lambda_{1:T} p(\mathbf{x}, \lambda_{1:T}|\mathbf{r}) \quad (2.5)$$

The joint conditional distribution  $p(\mathbf{x}, \lambda_{1:T}|\mathbf{r})$  can be decomposed into

$$p(\mathbf{x}, \lambda_{1:T}|\mathbf{r}) = p(\mathbf{x}|\lambda_{1:T}, \mathbf{r}) p(\lambda_{1:T}|\mathbf{r}) \quad (2.6)$$

Due to independence of hidden sequence  $\lambda_{1:T}$  and the alignment variables  $\mathbf{r}$ , the equation in Equation 2.6 can be further simplified into

$$\begin{aligned} p(\mathbf{x}, \lambda_{1:T}|\mathbf{r}) &= p(\mathbf{x}|\lambda_{1:T}, \mathbf{r}) p(\lambda_{1:T}) \\ &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} p(x_{k,n}|r_k, \lambda_{1:T}) \prod_{\tau=1}^T p(\lambda_{\tau}) \end{aligned} \quad (2.7)$$

Note that, such a decomposition of the joint distribution in Equation 2.5 is possible because  $\lambda_\tau$  are independent from each other (see Equation 2.3) and  $x_{k,n}$  are conditionally independent given  $\lambda_{1:T}$  and  $r_{1:K}$ . Combining Equations 2.5 and 2.7, one can obtain the likelihood distribution as follows:

$$p(\mathbf{x}|\mathbf{r}) = \int d\lambda_{1:T} \prod_{k=1}^K \prod_{n=0}^{N_k-1} p(x_{k,n}|r_k, \lambda_{1:T}) \prod_{\tau=1}^T p(\lambda_\tau) \quad (2.8)$$

One can achieve the optimum alignment  $r_{1:K}^*$  by computing the following:

$$r_{1:K}^* = \arg \max_{r_{1:K}} \Phi(r_{1:K}) \quad (2.9)$$

Note that scoring functions  $\Phi(\mathbf{r})$  are defined directly for arbitrary number of sequences including non-overlapping cases.<sup>4</sup>

This formulation in Equation 2.9 can also be interpreted from a Bayesian model selection perspective [36]. We are comparing the scores for different alignments i.e.,  $\Phi(\mathbf{r})$ , to find the 'best model' that describes the data where each configuration  $\mathbf{r}$  corresponds to a different model.

Note that this template model is quite flexible such that various feature representations can be modelled with the choice of an appropriate observation model and the score functions. To this end, we define six different generative models for positive, non-negative, real and binary features. The models follow the template generative model but with different choices of prior and observation models. These are listed in Table 2.1<sup>5</sup> and each generative model is briefly discussed in Section 2.1.

It is important to mention that by choosing prior and likelihood distributions as conjugate pairs, i.e., Gamma-Inverse Gamma, Bernoulli-Bernoulli, analytical derivation of the scoring function  $\Phi(\mathbf{r})$  is possible. The derivations of score functions for each

---

<sup>4</sup> $\Phi(\mathbf{r})$  functions also provide comparable scores for the alignments when the sequences do not overlap with each other.

<sup>5</sup>The exponential forms of prior and likelihood models for each generative model are given in Appendix A

generative model can be found in Appendix A.

## 2.1. Generative Models

### 2.1.1. Gamma observation model ( $M_1$ )

We use this model for positive and non-negative feature sets. In this work, we investigate two non-negative features: ( $F_1$ ) the energy in sub-bands in STFT, ( $F_2$ ) positive spectral difference [33]. We assume

$$\begin{aligned}\lambda_\tau &\sim \mathcal{IG}(\lambda_\tau; \alpha_\lambda, \beta_\lambda) \\ x_{k,n} &\sim \mathcal{G}(x_{k,n}; \alpha, \frac{\alpha}{\lambda_\tau})\end{aligned}$$

Here, hidden parameters  $\lambda_\tau$  are positive and inverse gamma distributed as a conjugate prior. Note that the conditional mean and the variance of the observation  $x_{k,n}$  are  $\lambda_\tau$  and  $\lambda_\tau^2/\alpha$  respectively. Here,  $\alpha$  is a precision parameter that controls and adjusts how much  $x_{k,n}$  deviates from the hidden parameter  $\lambda_\tau$ , i.e., as  $\alpha$  increases, the observed sequences should be more similar to the hidden sequence hence there is less deviations or less noise. On the other hand, smaller  $\lambda_\tau$  also increases the precision. Then choosing parameter  $\beta_\lambda$  smaller which decreases the mean and the variance of the hidden sequence, increases precision. Choosing parameter  $\alpha_\lambda$  higher also decreases the variance of hidden sequence hence increases precision of the model.

Table 2.1. Prior and likelihood distributions for each model.

Models	$p(\lambda_\tau)$	$p(x_{k,n} r_k, \lambda_\tau)$
Model 1 ( $M_1$ )	$\mathcal{IG}(\lambda_\tau; \alpha_\lambda, \beta_\lambda)$	$\mathcal{G}(x_{k,n}; \alpha, \frac{\alpha}{\lambda_\tau})$
Model 2 ( $M_2$ )	$\mathcal{IG}(\lambda_\tau; \alpha_\lambda, \beta_\lambda)$	$\mathcal{N}(x_{k,n}; 0, \lambda_\tau)$
Model 3 ( $M_3$ )	$\mathcal{N}(\lambda_\tau; 0, \beta_\lambda)$	$\mathcal{N}(x_{k,n}; \lambda_\tau, 1/\alpha)$
Model 4 ( $M_4$ )	$\mathcal{BE}(\lambda_\tau; \alpha_\lambda, \beta_\lambda)$	$\mathcal{BE}(x_{k,n}; \lambda_\tau)$
Model 5 ( $M_5$ )	$\mathcal{BE}(\lambda_\tau; \alpha_\lambda)$	$\mathcal{P}(x_{k,n}; \lambda_\tau, \mathbf{w})$
Model 6 ( $M_6$ )	$\text{Dir}(\lambda_{1:Q,\tau}; \alpha_{1:Q})$	$\mathcal{M}(x_{1:Q,k,n}; 1, \lambda_{1:Q,\tau})$

### 2.1.2. Gaussian variance observation model ( $M_2$ )

We use this model for real valued feature sets. In this work, we investigate three real features: ( $F_3$ ) MFCC, ( $F_6$ ) the first difference through time in STFT and ( $F_7$ ) the first difference through time and frequency in STFT. We assume

$$\begin{aligned}\lambda_\tau &\sim \mathcal{IG}(\lambda_\tau; \alpha_\lambda, \beta_\lambda) \\ x_{k,n} &\sim \mathcal{N}(x_{k,n}; 0, \lambda_\tau)\end{aligned}$$

Here, the hidden parameters  $\lambda_\tau$  act as the variance of the observation model hence they are positive and represented with inverse gamma distribution. Here, there is no explicit precision parameter,  $\lambda_\tau$  that directly determines how much  $x_{k,n}$  deviates from zero. Therefore precision of the model can be adjusted only by tuning the parameters of the hidden sequence. To increase precision of the model,  $\beta_\lambda$  is chosen small which decreases the mean and the variance of the hidden sequence and increases precision. Choosing  $\alpha_\lambda$  higher also decreases the variance of hidden sequence hence increases precision of the model.

### 2.1.3. Gaussian mean observation model ( $M_3$ )

We use this model again for real valued feature sets. As in the previous model, we investigate the feature sets  $F_3$ ,  $F_4$  and  $F_6$ . We assume

$$\begin{aligned}\lambda_\tau &\sim \mathcal{N}(\lambda_\tau; 0, \beta_\lambda) \\ x_{k,n} &\sim \mathcal{N}(x_{k,n}; \lambda_\tau, 1/\alpha)\end{aligned}$$

Here, the hidden parameters  $\lambda_\tau$  are real valued and Gaussian distributed. The mean and the variance of the observation  $x_{k,n}$  are  $\lambda_\tau$  and  $1/\alpha$ , respectively. As in  $M_1$ ,  $\alpha$  is a precision parameter that adjusts the deviation of  $x_{k,n}$  from  $\lambda_\tau$ , i.e., as  $\alpha$  increases, the observed sequences should be more similar to the hidden sequence hence the deviation or noise is less. Note that the observed data  $x_{k,n}$  has a mean value of  $\lambda_\tau$ , therefore the

variance parameter  $\beta_\lambda$  should be chosen accordingly so that the range of  $\lambda_\tau$  fits the observed data.

#### 2.1.4. Bernoulli observation model ( $M_4$ )

We use this model for binary feature sets. In this work, we investigate two binary features: ( $F_5$ ) thresholding the first difference through time and frequency in STFT [15] and ( $F_7$ ) thresholding the energy in sub-bands in STFT. We assume

$$\begin{aligned}\lambda_\tau &\sim \mathcal{B}(\lambda_\tau; \alpha_\lambda, \beta_\lambda) \\ x_{k,n} &\sim \mathcal{BE}(x_{k,n}; \lambda_\tau)\end{aligned}$$

Here, the hidden parameters are positive and represented with a beta distribution. The mean and variance of the observation  $x_{k,n}$  are  $\lambda_\tau$  and  $\lambda_\tau(1 - \lambda_\tau)$  respectively. As  $\lambda_\tau$  becomes closer to 1 or 0, the variance of observation also decreases hence the observed sequences should be more similar to hidden sequence and noise should be less. Setting of the hidden sequence parameters  $\alpha_\lambda$  and  $\beta_\lambda$  depends on the choice of feature set of the observations. For example, thresholding the energy in the sub-bands ( $F_7$ ) usually produces a sparse data (more 0's than 1's). In this case  $\alpha_\lambda$  should be much higher than  $\beta_\lambda$  ( $\alpha_\lambda \gg \beta_\lambda$ ). However for  $F_5$ , the number of 1's and 0's are almost equal. The precision of the model indirectly increases for small and equal values parameters.

#### 2.1.5. Conditional Bernoulli observation model ( $M_5$ )

We use this model for binary feature sets. As in  $M_4$ , we investigate the features  $F_5$  and  $F_7$ . We assume

$$\begin{aligned}\lambda_\tau &\sim \mathcal{BE}(\lambda_\tau; \alpha_\lambda) \\ x_{k,n} &\sim \mathcal{P}(x_{k,n}; \lambda_\tau, \mathbf{w})\end{aligned}$$

Here, the hidden parameters are binary and represented with a Bernoulli distribution. The observations are represented with a conditional Bernoulli distribution that is defined as

$$\mathcal{P}(x_{k,n}; \lambda_\tau, \mathbf{w}) = \exp\left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{k,n} = i][\lambda_\tau = j] \log(w_{i,j})\right)$$

where  $\mathbf{w}$  represents the parameter set  $\{w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1}\}$ . In this notation,  $w_{i,j}$  is defined as the probability that  $\lambda_\tau = j$  and  $x_{k,n} = i$ . We call it a 'conditional' Bernoulli distribution because for different values of hidden parameter  $\lambda_\tau$ , it becomes a Bernoulli distribution with different parameter, such as;

$$\mathcal{P}(x_{k,n}; \lambda_\tau, \mathbf{w}) = \begin{cases} \mathcal{BE}(x_{k,n}; w_{1,0}), & \lambda_\tau = 0 \\ \mathcal{BE}(x_{k,n}; w_{1,1}), & \lambda_\tau = 1 \end{cases}$$

We further assume that  $w_{0,0} = w_{1,1} = w$  and  $w_{1,0} = w_{0,1} = 1 - w$ . Note that  $w$  determines the amount of similarity between the observed data and the hidden parameter. Hence as  $w$  becomes closer to 1, the observed sequences should be more similar to hidden sequence. For  $F_5$  feature, we assume  $p(\lambda_\tau = 1) = p(\lambda_\tau = 0)$  so the hyperparameter of the hidden parameter is chosen as  $\alpha_\lambda = 0.5$ . On the other hand, for  $F_7$  feature where  $p(\lambda_\tau = 0) > p(\lambda_\tau = 1)$ ,  $\alpha_\lambda$  should be chosen closer to zero.

### 2.1.6. Multinomial observation model ( $M_6$ )

This model is an extended version of the model 4 where there are more than just two distinct levels hence it is useful when the features are categorical. Accordingly, we investigate the categorical feature: ( $F_8$ ) quantized spectral energy coefficients with  $Q$  levels. We assume

$$\lambda_{1:Q,\tau} \sim \text{Dir}(\lambda_{1:Q,\tau}; \alpha_{1:Q})$$

$$x_{1:Q,k,n} \sim \mathcal{M}(x_{1:Q,k,n}; \lambda_{1:Q,\tau})$$

Note that the multinomial distribution has the number of trial parameter as 1. Then the  $x_{1:Q,n,k}$  is a vector for which only one element of the vector is active and the rest of the elements are equal to zero. As an example, if there are  $Q = 3$  levels and the second level is selected, the vector is,  $x_{1:Q,n,k} = \{0, 1, 0\}$ . The parameters of hidden sequence  $\alpha_{1:Q}$  should be chosen in accordance with the observed data.

More detail on the feature extraction and selected features is given in Section 5.2. Note that we denote each scoring function with the associated model number as a subindex. As an example, for the gamma observation model ( $M_1$ ), we denote the derived scoring function as  $\Phi_{M_1}(\mathbf{r})$ .

### 3. PAIRWISE AUDIO SEQUENCE ALIGNMENT

In this chapter, we focus on pairwise alignment. We compare a subset of the derived scoring functions with deterministic similarity measures specifically correlation and Hamming distance to gain intuition about the score functions specifically  $\Phi_{M_3}(\mathbf{r})$ ,  $\Phi_{M_5}(\mathbf{r})$  and an alternative interpretation of  $M_5$  for which the score function is denoted as  $\Phi_{M_{5.1}}(\mathbf{r})$ . We also discuss the effect of hyperparameters on the score function behaviours.

A straightforward method to align two sequences is to compute the cross correlation of sequences for all relative shifts, and the best alignment is chosen as the shift where the correlation is maximum. In general, any similarity measure  $S_{\mathbf{x}_i, \mathbf{x}_j}$  that quantifies the similarity between two sequences  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , can be applied in the same way. Note that in pairwise cases, computing a similarity function  $S_{\mathbf{x}_i, \mathbf{x}_j}$  for all relative shifts is equivalent to freezing the first sequence i.e., fixing  $r_1$ , and computing  $S_{\mathbf{x}_i, \mathbf{x}_j}$  over all possible alignments of second sequence ( $r_2$ ). Hence for pairwise cases it becomes a 1-dimensional search i.e.,  $S_{\mathbf{x}_i, \mathbf{x}_j}(r_{1:2}) = S_{\mathbf{x}_i, \mathbf{x}_j}(r)$  since it is only a function of  $r_2$  (For simplicity, we use  $r$  instead of  $r_2$ ).

If the sequences are the recordings of the same auditory scene and they are overlapping on a common time line, the similarity measure would likely result in a distinctive score (maximum value) at the true alignment (shift)  $r$  so the optimum alignment  $r^*$  is computed as,

$$r^* = \arg \max_r S_{\mathbf{x}_i, \mathbf{x}_j}(r) \quad (3.1)$$

where  $S_{\mathbf{x}_i, \mathbf{x}_j}(r)$  represents any similarity measure between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ .

Although some of the similarity measures have a high alignment performance<sup>6</sup>, there are two major drawbacks with these methods in the alignment context:

- They can be used to align pairs of sequences, even if there are more than two sequences, the total alignment of sequences can be achieved by only applying these measures for all possible combinations of two sequences.
- It is not clear how to apply such similarity measures when the sequences are **not** overlapping. Note that these measures can not quantify the similarity of sequences if they are not overlapping on the time line.

By overlap, we mean that some parts of sequences are actually aligned at the same global time index as shown for  $x_1, x_2$  and  $x_3$  in Figure 2.1. For deciding overlap and non-overlap, arbitrary threshold parameters need to be selected. However such choices are very difficult to make optimally as they depend highly on data.

Our derived scoring functions  $\Phi(\mathbf{r})$  can also be interpreted as similarity measures. However in contrast to simple similarity measures,  $\Phi(\mathbf{r})$  functions give sensible and comparable scores for all relative shifts including the non-overlap cases. Again applying the same strategy, scores for all possible shifts are computed and the shift with the highest score is decided as the optimum alignment. A toy example is shown in Figure 3.1 where pairwise alignment procedure is revealed. There are two sequences  $x_1$  and  $x_2$  with lengths  $N_1 = 4$  and  $N_2 = 3$ , respectively. One sequence is freezed at time  $\tau = N_2 + 1 = 4$  ( $r_1 = 4$ ), and the score function is  $\Phi(r)$  is computed for all possible shifts for second sequence  $r$ . Note that non-overlapping alignment score is computed for  $r = 1$ . It is also important to mention that in the pairwise case, for any relative shift that leads to a non-overlapping case between sequences,  $\Phi(r)$  function results in the same score. Hence for the pairwise case, computing  $\Phi(r)$  for just one shift where the sequences do not overlap would be enough to handle non-overlapping cases.

---

<sup>6</sup>Specifically cross correlation and Hamming Distance measures are tested against pairwise data sets with various types and amounts of noise in Section 5.4

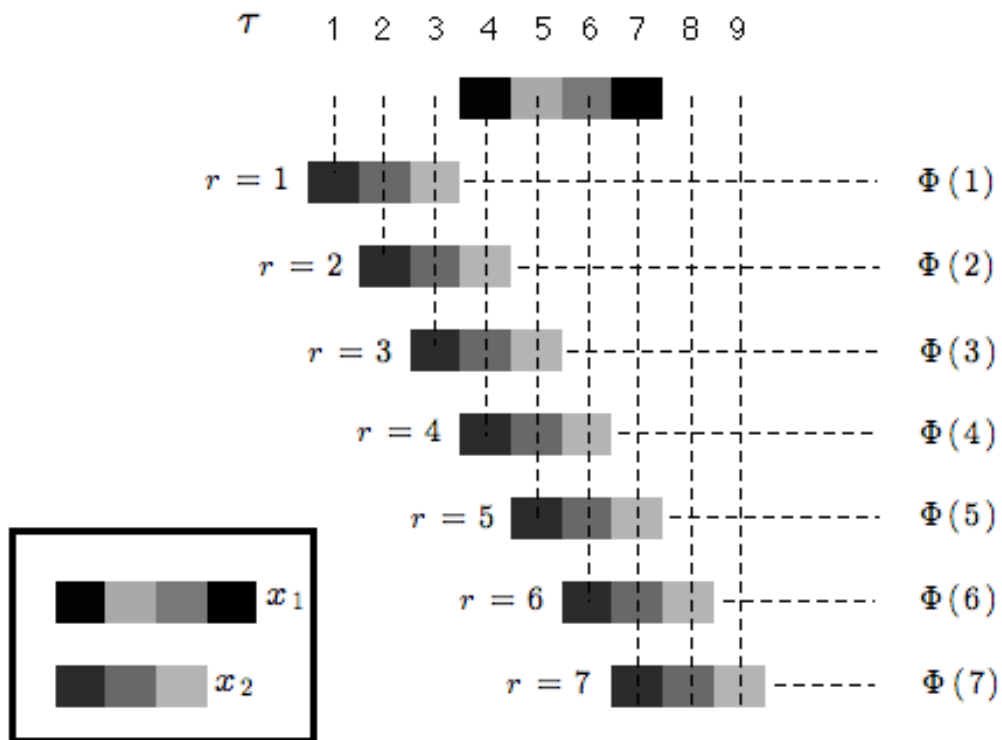


Figure 3.1. Illustration of pairwise alignment with a toy example.  $x_1$  and  $x_2$  are two sequences to be aligned (inside the rectangle).  $\tau$  is the global time index and  $r$  is the current alignment of  $x_2$ .  $\Phi(r)$  is shown for all relative shifts.

As we mentioned,  $\Phi(r)$  functions behave like similarity measures thus we compare a subset of them with simple similarity measures in particular cross correlation and Hamming distance analytically in Section 3.1 and Section 3.2, respectively. Then we compare the alignment performances of these methods under different alignment scenarios in Section 3.3.

### 3.1. Relation between cross correlation and Gaussian mean observation model ( $M_3$ )

In this section, we investigate the relationship between one of the most common similarity measures namely cross correlation and the score function  $\Phi_{M_3}(\mathbf{r})$  that is derived from Gaussian mean observation model  $M_3$ . Cross correlation is a well-known similarity measure that is used in various different fields such as statistics, economics and audio signal processing. The popularity of this method is due to two reasons;

- It is easy to compute (Fast Fourier Transform (FFT) methods)
- It can be applied on transform domains as well as time domain (generalized cross-correlation) [29]

The cross correlation of two sequences is obtained as,

$$R_{x_1, x_2}(r) = \sum_{\tau=-\infty}^{\infty} x_1(\tau)x_2(r + \tau) \quad (3.2)$$

The relationship between  $\Phi_{M_3}(r)$  and cross correlation measures can be obtained by analyzing the structure of  $\Phi_{M_3}(r)$  formulation<sup>7</sup>. For pairwise cases,  $\Phi_{M_3}(r)$  simplifies into,

$$\Phi_{M_3}(r) = A(\mathbf{x}, \beta_\lambda, \alpha, r) + \frac{1}{\beta_\lambda(\beta_\lambda\alpha + 2)} \sum_{\tau=1}^T x_{1, \tau-N_2+1}x_{2, \tau-r} \quad (3.3)$$

$\Phi_{M_3}(r)$  is decomposed into two parts in Equation 3.3 where the second term actually computes the correlation between two sequences for an alignment  $r$ . Note that the effect of second term (correlation term) in the equation is tuned with the choice of hyperparameters  $\beta_\lambda$  and  $\alpha$ . We denote the rest of the formula with  $A(\mathbf{x}, \beta_\lambda, \alpha, r)$ .

---

<sup>7</sup>The derivation of  $\Phi_{M_3}(r)$  is given in Appendix A.3 for general and pairwise cases.

To further analyze  $\Phi_{M_3}(r)$ ,  $A(\mathbf{x}, \beta_\lambda, \alpha, r)$  part is decomposed into subparts as,

$$A(\mathbf{x}, \beta_\lambda, \alpha, r) = A_0(\mathbf{x}, \beta_\lambda, \alpha) + A_1(\mathbf{x}, \beta_\lambda, \alpha, r) + A_2(\mathbf{x}, \beta_\lambda, \alpha, r) \quad (3.4)$$

where

$$\begin{aligned} A_0(\mathbf{x}, \beta_\lambda, \alpha) &= -\frac{TF}{2} \log(2\pi\beta_\lambda) - TF(N_1 + N_2)\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \\ A_1(\mathbf{x}, \beta_\lambda, \alpha, r) &= \frac{\alpha}{2} \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_2+1} [n = \tau - r])} - 1 \right) \sum_{f=1}^F x_{1,\tau-N_2-1,f}^2 \\ A_2(\mathbf{x}, \beta_\lambda, \alpha, r) &= \frac{\alpha}{2} \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_1+1} [n = \tau - N_2 - 1])} - 1 \right) \sum_{f=1}^F x_{2,\tau-r,f}^2 \end{aligned}$$

In Equation 3.4, the  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  parts have a similar structure. In  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$ , the square of each feature coefficient  $x_{1,n,f}$  is scaled with a coefficient and summed. The scaling coefficients always take negative values and if a coefficient is overlapping with a coefficient of second sequence  $x_2$ , then the absolute value of the scaling becomes larger thus resulting sum becomes smaller. A similar effect is observed in  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$ . Thus, we conclude that for the alignments that results in high overlap between sequences,  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  would have lower scores and for alignments that result in less overlap between sequences,  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  would have higher scores. The correlation term would have a distinctive score at the true alignment (if the sequences are actually overlapping at the true alignment). Note that the  $A_0(\mathbf{x}, \beta_\lambda, \alpha)$  does not depend on the alignment  $r$ , hence it is constant for all alignments.

A simple alignment example is shown in Figure 3.2 where the effect of each part  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  are analyzed as well as the correlation term. In this example, there are two sequences that overlap with each other on the time line. The feature  $F_4$  is used in the feature extraction and  $\Phi_{M_3}(r)$  for all possible alignments  $r$  are computed.  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  are also computed for all  $r$  and shown in the Figure 3.2 as separate plots. It can be observed that the parts  $A_1$  and  $A_2$  have higher scores for less overlapping shifts and lower scores for more overlapping shifts. The correlation term has a distinctive peak at the true alignment (shift) of sequences.

Then the  $\Phi_{M_3}(r)$  function is computed by summing all these parts.

If the sequences are overlapping at the true alignment, the effect of score in correlation term has to overcome the decreasing effect of  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  to be able to have a global peak in  $\Phi_{M_3}(r)$  at the true alignment. On the other hand, if the sequences are not overlapping, then the correlation would not have a distinctive score. In this case, the combined effect of  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$  and  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$  has to overcome the effect of correlation so that the global peak occurs at the sides of  $\Phi_{M_3}(r)$ .

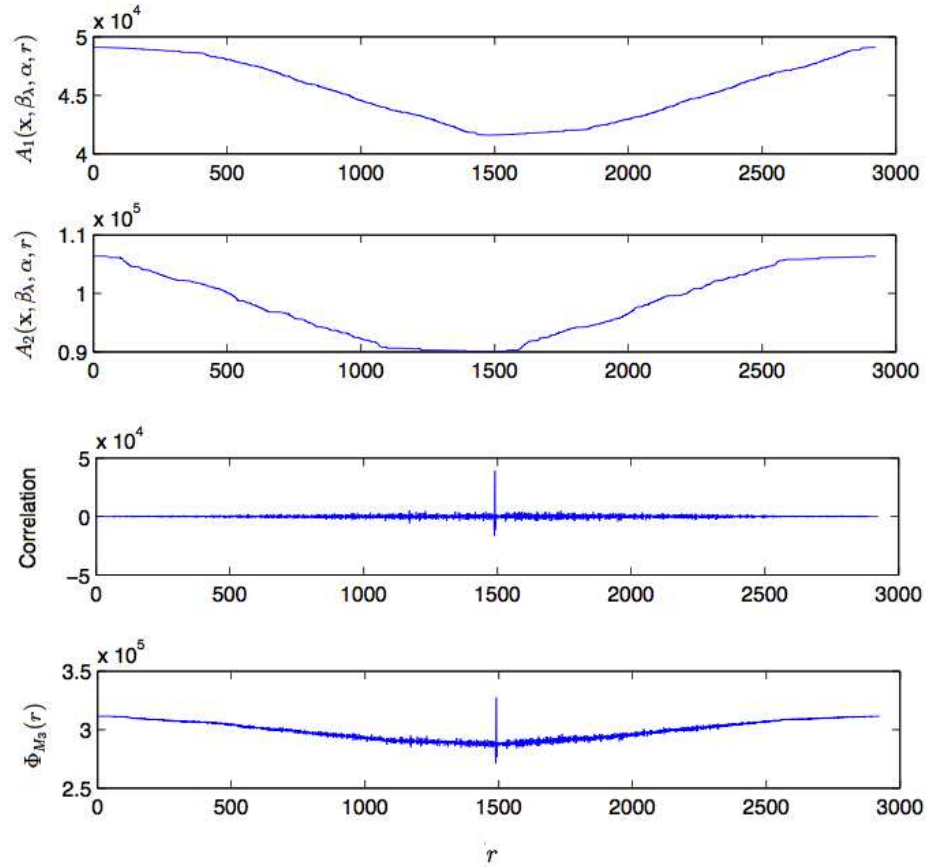


Figure 3.2. Subparts of  $\Phi_{M_3}(r)$  for two overlapping sequences, from top to bottom  $A_1(\mathbf{x}, \beta_\lambda, \alpha, r)$ ,  $A_2(\mathbf{x}, \beta_\lambda, \alpha, r)$ , correlation term and resulting  $\Phi_{M_3}(r)$ .

### 3.2. Relation between Hamming distance and Conditional Bernoulli observation model ( $M_5$ )

A similar analogy exists between Hamming distance and conditional Bernoulli observation model ( $M_5$ ). Hamming distance between two sequences of equal length is defined as the number of positions at which the corresponding symbols are different or the minimum number of errors that could have transformed one string into the other [37]. When the feature sequences are binary, the normalized Hamming distance can be used to quantify the similarity between the overlapping parts of the sequences i.e., a bitwise comparison in the overlapping parts of the signals can be used as a similarity measure.

In Table 3.1, an example of such a situation is shown. If two coefficients of sources  $x_1$  and  $x_2$  i.e.,  $x_{1,1,1}$  and  $x_{2,0,1}$  that are aligned to the time  $\tau = 1$ , are equal to each other then they are counted as 1, otherwise they are not counted (Hamming distance). The ratio of this count to the total number of overlapping bits (normalized Hamming distance) acts as a similarity measure since at the exact alignment, this ratio should be highest. In this scenario, there are 6 overlapping bits and 4 of them are equal to each other therefore the ratio is computed as  $2/3$ . Note that the ratio of overlapping and equal number of bits to the total number of aligned number of bits is a more suitable measure in this case because otherwise it will not give comparable distance results for different alignments.

Since we are using a ratio as a measure, as the overlap between sequences increases, the score becomes more accurate. For the alignments where the overlap is very small, the ratio could be high by chance, therefore the similarity scores for the alignments where the overlap is lower, are not so reliable. An example of the similarity score that is computed between two overlapping sequences for all possible alignments is shown in Figure 3.3 . It can be observed that there is a significant peak at the true alignment point with a ratio of around 0.7.

To investigate the relation between Hamming distance and the  $\Phi_{M_5}(r)$  function

Table 3.1. Similarity of two binary sequences  $x_{1,0:3,1:2}$  and  $x_{2,0:3,1:2}$  aligned at  $r_1 = 2$  and  $r_2 = 3$ .

$\tau =$	1	2	3	4	5	6
$r_1 = 2$		$x_{1,0,1} = 1$ $x_{1,0,2} = 0$	$x_{1,1,1} = 1$ $x_{1,1,2} = 1$	$x_{1,2,1} = 0$ $x_{1,2,2} = 1$	$x_{1,2,1} = 1$ $x_{1,2,2} = 0$	
$r_2 = 3$			$x_{2,0,1} = 1$ $x_{2,0,2} = 1$	$x_{2,1,1} = 0$ $x_{2,1,2} = 0$	$x_{2,2,1} = 1$ $x_{2,2,2} = 1$	$x_{2,3,1} = 0$ $x_{2,3,2} = 0$

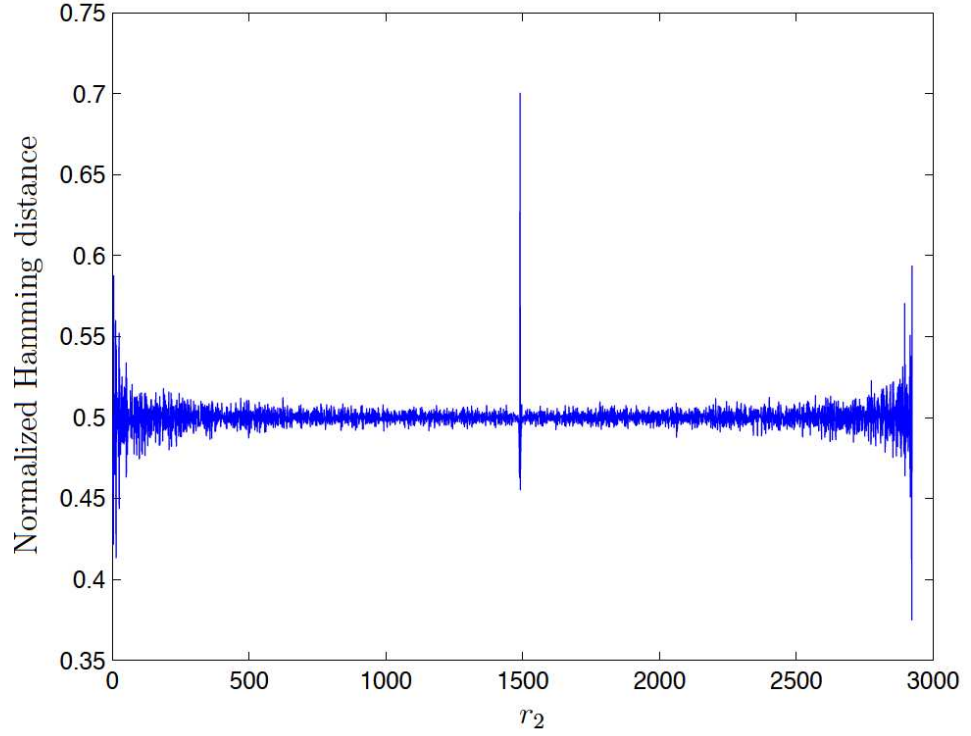


Figure 3.3. Normalized Hamming distance for all relative shifts  $r_2$  between two overlapping sequences.

that is derived from conditional Bernoulli observation model, one needs to analyze the structure of the  $\Phi_{M_5}(r)$  formulation<sup>8</sup>. For pairwise cases,  $\Phi_{M_5}(r)$  simplifies into,

<sup>8</sup>The derivation of  $\Phi_{M_5}(r)$  is given in Appendix A.5 for general and pairwise cases.

$$\Phi_{M_5}(\mathbf{r}) = \sum_{\tau=1}^T \log \left( 0.5(1-w)^{([x_{1,\tau-r_1}=0]+[x_{2,\tau-r_2}=0])} w^{([x_{1,\tau-r_1}=1]+[x_{2,\tau-r_2}=1])} + 0.5(1-w)^{([x_{1,\tau-r_1}=1]+[x_{2,\tau-r_2}=1])} w^{([x_{1,\tau-r_1}=0]+[x_{2,\tau-r_2}=0])} \right) \quad (3.5)$$

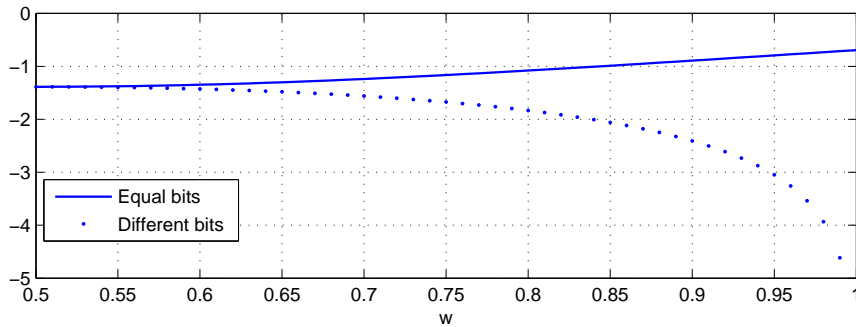


Figure 3.4. The contribution of aligned equal bits and aligned different bits versus  $w$  parameter on computation of  $\Phi_{M_5}(r)$  score function.

Since the model is defined for binary sequences, two aligned coefficients of sequences are either equal to each other or not (both 0's or 1's). When two aligned coefficients are equal to each other, the contribution to the sum in Equation 3.5 is  $\log(0.5(1-w)^2 + 0.5w^2)$ , whereas when two aligned coefficients are not equal, the contribution to the sum is  $\log(w(1-w))$ . In Figure 3.4, the contribution of aligned equal bits and aligned different bits to the  $\Phi_{M_5}(r)$  computation are shown for the varying  $w$  parameter in the range  $0.5 \leq w \leq 1$ . It can be observed that  $\Phi_{M_5}(r)$  would be higher when more bits are equal in a relative shift. This is equivalent to counting the number of equal and different bits in the overlapping region hence the relation to the Hamming distance.

Like cross correlation, Hamming distance can not be used to quantify the similarity between non-overlapping shifts between sequences. Hence  $\Phi_{M_5}(r)$  is a superior similarity measure to normalized Hamming distance measure in the alignment context because it benefits from the strength of the Hamming distance while overcoming its drawback.

### 3.3. Comparison between correlation and Hamming distance alignment performances with $\Phi_{M_3}(\mathbf{r})$ and $\Phi_{M_5}(\mathbf{r})$ scoring function performances under different scenarios

To compare the performances of cross-correlation,  $R(\mathbf{r})$ , and normalized Hamming distance with the associated scoring functions  $\Phi_{M_3}(\mathbf{r})$  and  $\Phi_{M_5}(\mathbf{r})$  under different conditions, we compute  $R(\mathbf{r})$  and  $\Phi_{M_3}(\mathbf{r})$  with feature  $F_4$ ,  $H(\mathbf{r})$  and  $\Phi_{M_5}(\mathbf{r})$  with feature  $F_5$  for all relative shifts in three different cases (all pairwise) and the results are plotted in Figure 3.5. Here we define  $H(\mathbf{r})$  as '1-normalized Hamming distance' and we compute  $H(\mathbf{r})$  instead of Hamming distance, in order to have a maximum rather than a minimum at the offset value for illustration purposes. Denoting the lengths of sequences as  $N_1$  and  $N_2$ , there are  $N_1 + N_2$  relative shifts to be computed. In the examples in Figure 3.5, we freeze  $r_1 = 0$  and compute  $\Phi(r)$  in the range  $r = -N_2 : N_1$  where the first relative shift ( $r = -N_2$ ) represents the shift where the sequences do not overlap.

In the extraction of features  $F_4$  and  $F_5$ , the STFT hop size is chosen as 20 ms hence each relative shift represents 20 ms. Note that the x-axis is the time in seconds in each plot in Figure 3.5 that is computed by multiplying each shift with 20 ms.

*Case 1 (Overlapping):* The sequences are overlapping on a common time line. The length of sequences are around 1 minute each and they have an offset of 4.32 seconds. The recordings consist of speech with a high signal to noise ratio (SNR) against the environmental noise (higher than 10dB). The resulting  $R(r)$ ,  $H(r)$ ,  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  are plotted in Figure 3.5(a), 3.5(d), 3.5(g) and 3.5(j) respectively (the left column in Figure 3.5). It can be observed that both similarity measures and scoring functions have distinctive peaks at the offset value.

*Case 2 (Non-overlapping):* The same procedure is applied for two non-overlapping sequences that are the recordings of a television show with high SNR. The sequences are of length 1 minute each. The resulting  $R(r)$ ,  $H(r)$ ,  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  are plotted in Figure 3.5(b), 3.5(e), 3.5(h) and 3.5(k) respectively (the middle column in Figure

3.5). Note that the first value in both  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  represents the case (shift) where the sequences do not overlap which solves overlap/non-overlap issue. On the other hand,  $R(r)$  and  $H(r)$  are defined as zero for non-overlapping shifts hence are not able to quantify non-overlapping cases.

*Case 3 (Noisy):* We compare these functions in a difficult scenario. By 'difficult' we mean that the sequences are highly contaminated with noise (wind noise in this case). The first sequence is of length 5.5 minutes (335 seconds) and the second sequence is of length 56 seconds, and they have an offset of 230.32 seconds. The results are plotted for  $R(r)$ ,  $H(r)$ ,  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  in Figure 3.5(c), 3.5(f), 3.5(i) and 3.5(l) respectively (the right column in Figure 3.5). It is observed that both  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  have distinctive peaks at the true offset. On the other hand, although  $R(r)$  and  $H(r)$  have peaks at the true offset, the spurious peaks at the sides of  $R(r)$  and  $H(r)$  are higher than the true peaks for both functions.

As a result of these cases, we observe that  $\Phi(r)$  functions have distinctive peaks at the true offset of sequences that match with each other. Even under very noisy conditions where simple similarity measures fail,  $\Phi(r)$  functions are able to identify the true alignment. We also observe that there are spurious peaks at the sides of  $R(r)$  and  $H(r)$  in Figure 3.5(a)-3.5(f). This is due to unreliability of such distance measures for alignment when the data is not sufficient (short duration of overlaps).

Note that the score functions  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  have v-shape structures. This is due to the nature of Bayesian model selection. These functions score how well the data is predicted by the model. For a specific alignment  $r$  where there is no matching or a partial small overlap between sequences, the model predicts the data better because most of the latent variables ( $\lambda_\tau$ ) generate a single coefficient hence the likelihood will be higher. In the opposite case when the overlap between sequences increases, the prediction of data becomes worse because most of the latent variables generate more than a single coefficient which leads to decreasing the the likelihood. On the true lag between sequences (if they are overlapping), the prediction is so good that a peak occurs. In other words, the model tries not to match these two sequences so the score for

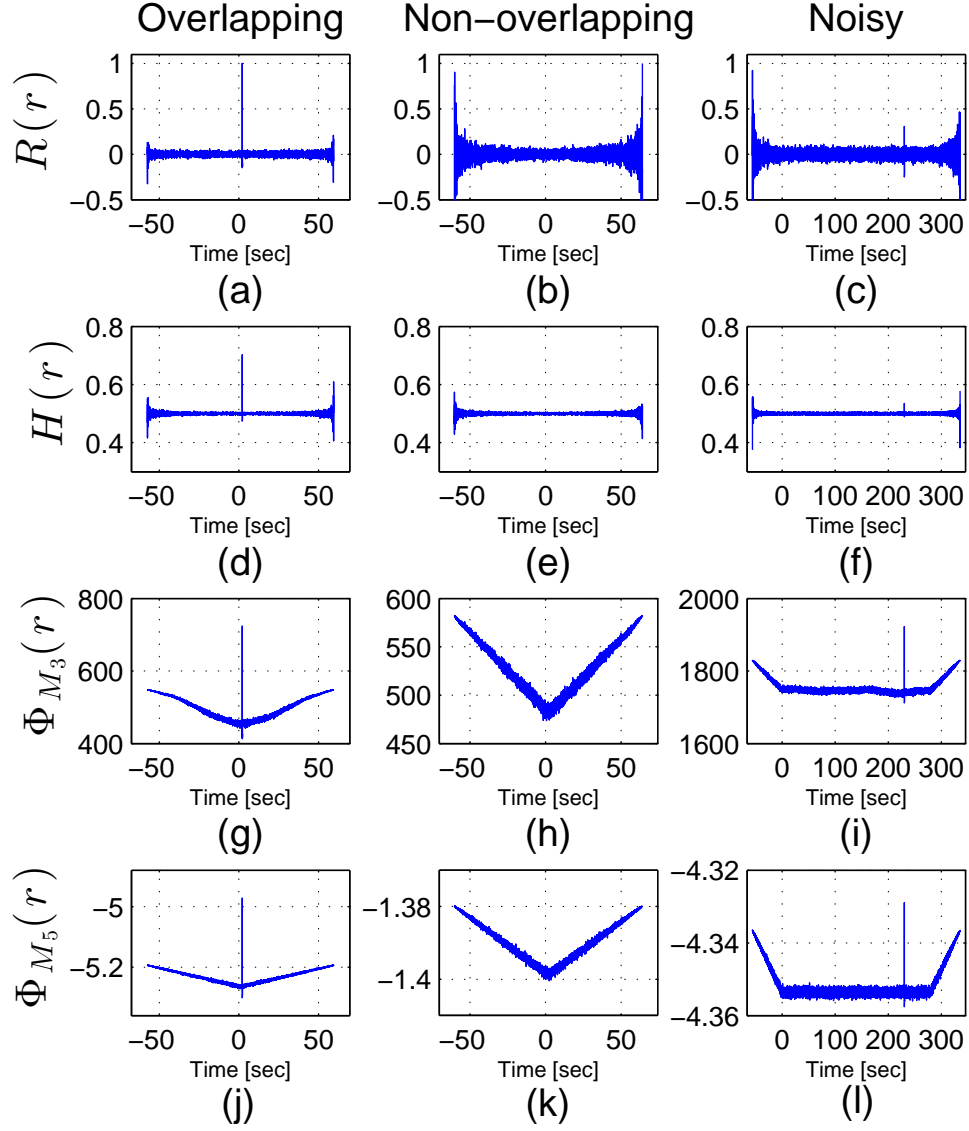


Figure 3.5. Pairwise alignments using  $R(r)$ ,  $H(r)$ ,  $\Phi_{M_3}(r)$  and  $\Phi_{M_5}(r)$  for three different scenarios. The left column (a,d,g,j): Two overlapping sequences (high SNR), the middle column (b,e,h,k): Two non-overlapping sequences (high SNR), the right column (c,f,i,l): Two overlapping sequences (Low SNR).

not overlapping or small overlapping parts (left and rightmost regions) is higher hence the 'v'-shape. This structure of score functions  $\Phi(\mathbf{r})$  is actually the main contribution of modelling approach. We do not need to specify some ad hoc threshold value as in the correlation, the model automatically tests if the sequences are not overlapping

or determines the true lag if they are overlapping. For pairwise alignment, it may be still feasible to determine a good threshold by trial and error for similarity measures. However in multi sequence alignment, it is increasingly harder to search and determine such thresholds.

### 3.4. Hyperparameter Choices

The correct choice of the hyperparameters  $\theta$  has a big impact on the alignment performance. This is due to the fact that the level of similarity between the observed sequences, is directly controlled with the precision parameter of the observation model. If we assume that the noise contamination or distortion is high for two overlapping sequences (low SNR cases i.e. smaller than -3dB), then it is likely that the sequences are not too similar at the overlapping parts. Then the precision of the observation model should be low enough (or variance of the model should be high enough) to let variability between aligned coefficients of sequences. On the other hand, when the aligned sequences are similar to each other (high SNR cases i.e. larger than 10dB), the precision of the observation model should be high enough (smaller variance) to let less variability between aligned coefficients of sequences thus the hyperparameters are chosen accordingly.

To illustrate this issue, an example is given for  $M_5$  model where the only hyperparameter is  $w$  which is proportional to the precision of the model<sup>9</sup> in Figure 3.6. In this example, there are two overlapping sequences and  $\Phi_{M_5}(r, w)$  is computed and plotted where the  $w$  parameter is changed linearly from 0.5 and 0.8. Note that in Section 2, we explain that by fixing the hyperparameters, we are able to drop the dependence on the hyperparameters i.e.,  $\Phi_{M_5}(r, w) = \Phi_{M_5}(r)$ . It can be observed that there is a distinctive peak in  $\Phi_{M_5}(r, w)$  function for each  $w$  value. However, for high  $w$  values (high precision), this peak stays below the non-overlapping alignment score. On the other hand, for small  $w$  values (low precision), that distinctive peak becomes the global peak of the function. This example illustrates the importance of hyperparameter choice on

---

<sup>9</sup>Actually there is another hyperparameter  $\alpha_\lambda$  in  $M_5$ , however, it is assumed  $\alpha_\lambda = 0.5$  in this example.

the performance of the alignment, because when the precision is too high i.e.,  $w > 0.8$ , it is not possible to find the true alignment of the sequences. Same is true when the precision is too low i.e.,  $0.5 < w < 0.55$  for the sequences that do not overlap in the time line. When the precision is too low, any spurious peak would be decided as the optimum alignment.

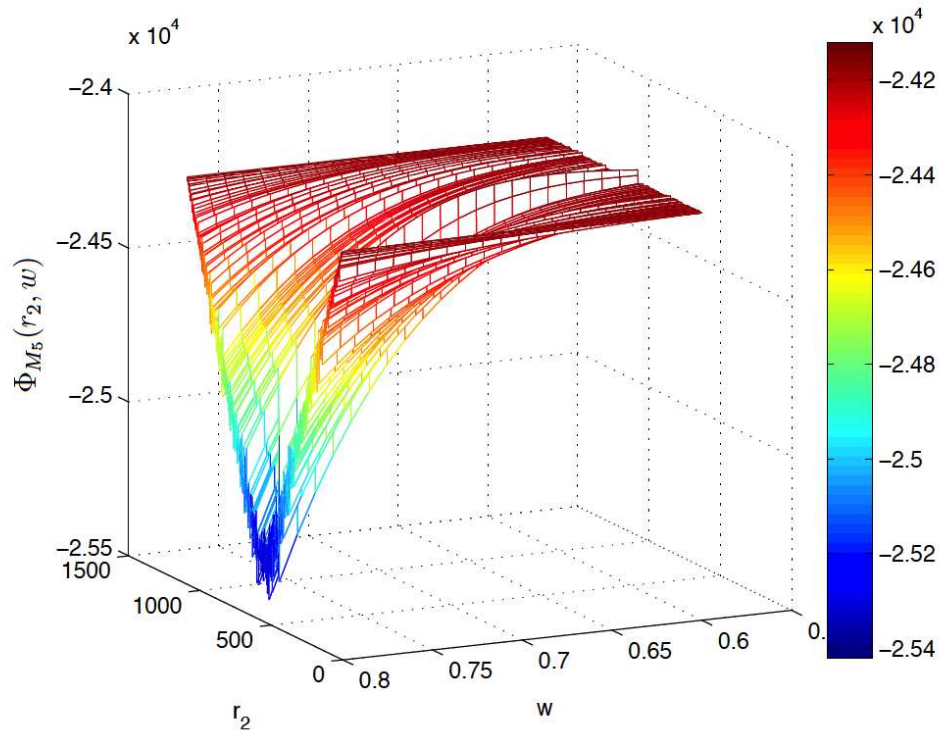


Figure 3.6. 3D plot of  $\Phi_{M_5}(r_2, w)$  over  $r_2$  and  $w$ .

Note that as the precision of the observation model decreases, some spurious peaks occur at the sides of corresponding  $\Phi(r)$  function. Thus, the choice of hyperparameters effect the reliability of  $\Phi(r)$  functions at the sides or small overlap cases. In Figure 3.7, an example of such a case is shown for  $\Phi_{M_5}(r)$  for two non-overlapping sequences.  $\Phi_{M_5}(r)$  is computed with both  $w = 0.75$  and  $w = 0.6$  and plotted in Figure 3.7(a) and 3.7(b), respectively. In addition, a zoomed version of  $\Phi_{M_5}(r)$  with  $w = 0.6$  is also plotted in Figure 3.7(c). The precision of  $M_5$  model is proportional with  $w$ , thus for smaller  $w$  in Figure 3.7(c) we observe spurious peaks at the sides.

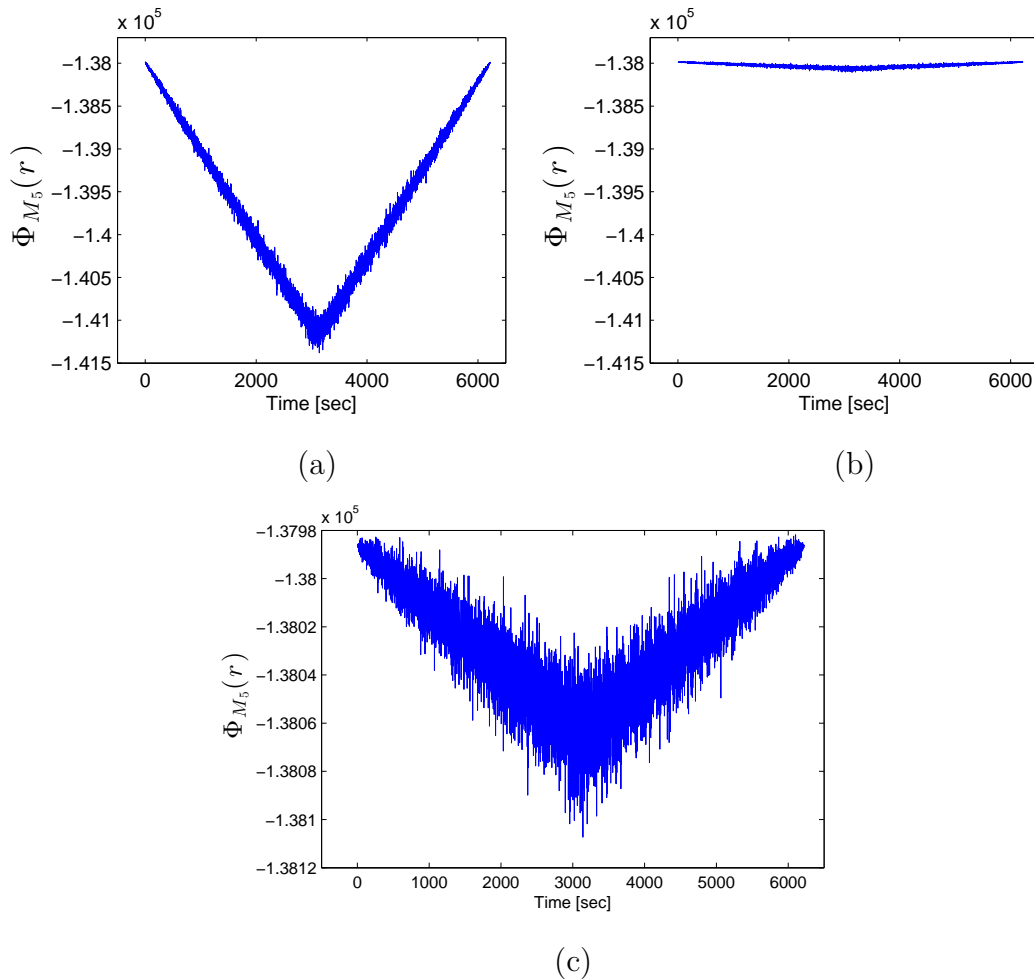


Figure 3.7.  $\Phi_{M_5}(r)$  results for two non-overlapping sequences computed for two different  $w$  parameters (a)  $\Phi_{M_5}(r)$ ,  $w = 0.75$  (b)  $\Phi_{M_5}(r)$ ,  $w = 0.6$  (c) Zoomed version of  $\Phi_{M_5}(r)$ ,  $w = 0.6$ .

For this reason, we define a '*reliable*' alignment as the case when the aligned sequences has an overlap region longer than some threshold  $\epsilon$  i.e.,  $> 1$  seconds, otherwise the alignment is assumed to be unreliable (small or no overlap). Note that the unreliable region becomes wider for larger variance (smaller precision).

A more detailed analysis on hyperparameter choices and estimation is given in Section 5.3.

### 3.5. Conditional Bernoulli Observation Model ( $M_5$ ) - Alternative Approach for Pairwise Alignment

In this section, we discuss an alternative interpretation of conditional Bernoulli observation model ( $M_5$ ) which is suitable only for pairwise alignment problems i.e., two observed sequences with lengths  $N_1$  and  $N_2$ . In our traditional strategy for alignment, we assume that these two sequences are noisy observations of another hidden(unobserved) sequence  $\lambda$ , fix the alignment of first sequence  $r_1 = N_2 + 1$  and search the optimum alignment of second sequence  $r^*$  over all possible lags between sequences including the alignment for non-overlap. However in this approach, we assume that one of the observations is actually the part of the hidden sequence  $\lambda$  that is not hidden anymore. Note that there are still unobserved parts of the hidden sequence. The other observed sequence is again a noisy realization of the hidden sequence. To be precise, the total length of the hidden sequence is assumed to be  $T = N_1 + 2 * N_2 - 1$ , the observed part of the hidden sequence is  $\lambda_{N_2+1:N_1+N_2,1:F}$  with length  $N_1$  and the other observed sequence is  $x_{0:N_2-1,1:F}$  with length  $N_2$ . The joint conditional distribution  $p(x_{0:N_2-1,1:F}|r)$  can be obtained by summing over the unobserved parts of  $\lambda_{\tau,f}$  i.e.,  $\lambda_{1:N_2,1:F}$  and  $\lambda_{N_1+N_2+1:T,1:F}$  such as,

$$p(x_{0:N_2-1,1:F}|r) = \sum_{\lambda_{1:N_2,1:F}} \sum_{\lambda_{N_1+N_2+1:T,1:F}} p(x_{2,0:N_2-1,1:F}, \lambda_{1:T,1:F}|r) \quad (3.6)$$

Note that there is only one sequence that has to be aligned to hidden part therefore in this case, we remove the sequence index  $k$  from  $x$  and  $r$  variables. The derivation of the score function which we denote as  $\Phi_{M_{5.1}}(r)$ , is given in Appendix A.6. The resulting score function is,

$$\begin{aligned}
\Phi_{M_{5.1}}(r) = & \sum_{\tau=N_2+1}^{N_1+N_2} \sum_{f=1}^F \left( ([x_{\tau-r,f} = 0][\lambda_{\tau,f} = 0] + [x_{\tau-r,f} = 1][\lambda_{\tau,f} = 1]) \log(1-w) \right. \\
& \left. + ([x_{\tau-r,f} = 0][\lambda_{\tau,f} = 1] + [x_{\tau-r,f} = 1][\lambda_{\tau,f} = 0]) \log(w) \right) \\
& + F \log(0.5) \left( \sum_{\tau=1}^{N_2} \mathcal{U}(r, \tau) + \sum_{\tau=N_1+N_2+1}^T \mathcal{U}(r, \tau) \right) \tag{3.7}
\end{aligned}$$

where

$$\mathcal{U}(r, \tau) = \begin{cases} 1 & \text{if } x_{0:N_2-1,1:F} \text{ exists at time } \tau; \\ 0 & \text{else.} \end{cases}$$

In the derivation of Equation 3.7, we assume the hyperparameters are  $\alpha_\lambda = 0.5$  and  $w_{0,0} = w_{1,1} = w$  and  $w_{0,1} = w_{1,0} = 1 - w$ . With this configuration,  $p(\lambda_{\tau,f} = 1) = p(\lambda_{\tau,f} = 0) = 0.5$  (1s and 0s are equiprobable) and the probability of a hidden coefficient  $\lambda_{\tau,f}$  and a coefficient that is aligned to time  $\tau$ ,  $x_{n,f}$  to be equal is  $w$  (Noisy observation  $x_{n,f}$  has the same value, 1 or 0, with the hidden coefficient  $\lambda_{\tau,f}$  with probability  $w$ ).  $\mathcal{U}(r, \tau)$  is a function of alignment and global time index that indicates if any coefficient of  $x_{0:N_2-1,1:F}$  is aligned at time  $\tau$  according to the alignment  $r$ .

Analyzing Equation 3.7, we observe that the first term covers the time interval  $N_2 + 1 : N_1 + N_2$  (the observed part of the hidden sequence) and the second term covers the rest of the time line  $1 : N_2$  and  $N_1 + N_2 + 1 : T$  (the non-overlapping parts of observed sequence). Here, the first term in the score function acts like the normalized Hamming distance similarity measure. If a coefficient of the observed sequence is equal to the corresponding coefficient in the observed part of  $\lambda_{\tau,f}$ , the contribution to the sum is a less negative number,  $\log(w)$ , rather than  $\log(1-w)$  (Assuming that  $w > 0.5$ ). This is equivalent to counting the coefficients that correspond to same time  $\tau$  (aligned at  $\tau$ ) and equal to each other (Hamming distance). Note that as the overlap between  $x_{0:N_2-1,1:F}$  and observed part of the hidden sequence increases, the contribution of the first term also increases hence a more negative value is obtained, that is why higher overlap results in smaller scores (v-shape structure, explained in Section 3.3). But only

for the true lag or alignment between sequences, this term gives a high value due to its behaviour similar to Hamming distance. The second term acts like a correction term for non-overlapping cases. It actually computes the number of coefficients of  $x_{0:N_2+1,1:F}$  that do not overlap with the observed part of the hidden sequence and multiplies with a negative number  $\log(0.5)$ . Therefore as the overlap decreases, the contribution of the second term increases. As a result, the second term gives higher scores for higher overlapping alignments balancing the effect of the first term.

One interesting issue is that in the Hamming distance similarity measure, the highest ratio which occurs at the true alignment point gives a good estimate of the hyper-parameter  $w$  for the model. It can be used as preprocessing step for computation of the hyper-parameter of the model. As an example, in Figure 3.3, the peak value at the true alignment is equal to 0.7. It can be interpreted as 70% of the data is equal to the corresponding hidden coefficient so the probability of being equal ( $p(\lambda_{\tau,f} = x_{n,f})$ ) is roughly 0.7.

This model is especially interesting for pairwise matching for two reasons,

- If the number of subbands is chosen as 32, each column of a sequence can be stored as 32-bit integer
- Since the computation is based on counting the number of similar bits in the overlapping parts of sequences, the likelihood can be computed very fast by using an XOR operation.

## 4. MULTIPLE AUDIO SEQUENCE ALIGNMENT

In this chapter, we discuss the alignment problem for multiple audio sequences. Having defined  $\Phi(\mathbf{r})$  as objective functions, multiple sequence alignment becomes a combinatorial optimization problem where we search the global maximum of cost function:  $r_{1:K}^* = \arg \max_{r_{1:K}} \Phi(r_{1:K})$ . When there are only two sequences (pairwise case), it is straightforward to compute  $\Phi(r)$  for all possible shifts (search all space) as in Section 3. However, when the number of sequences  $K$  is large, the search space grows exponentially in the order  $O(N^{K-1})$  where  $N$  is the average length of a sequence. Thus, searching the entire space of each possible alignment (each possible relative shift) is clearly intractable.

For such cases, standard optimization approach is to start from an initial point  $\hat{r}_{1:K}$  and to gradually move towards the global maximum of the cost function hence reach the maximum without visiting all the space. However the  $\Phi(\mathbf{r})$  surface is very rough and standard optimization methods such as gradient based methods are not applicable. For this reason, we resort to some sampling strategies and a heuristic sequential search algorithm. As our preliminary work, we proposed a Gibbs sampling algorithm in [38] where we sample from the joint conditional posterior distribution  $p(\lambda, \mathbf{r}|\mathbf{x})$ . Then a heuristic sequential search algorithm is proposed in [39] and improved in [40]. In [41], a multi resolution alignment algorithm using SMC samplers is proposed for pairwise cases. Here we extend the method for multiple sequence alignment scenarios.

### 4.1. Gibbs Sampling with Simulated Tempering

Gibbs sampling is one of the most popular Markov Chain Monte Carlo (MCMC) methods which is used to sample from distributions with at least two dimensions [42], [43]. Sampling methods are usually used in cases where the target distribution is too complex to do some analytical computation as in our situation. In [38], we use the Gibbs sampler to sample from the conditional joint distribution  $p(\lambda_{1:T}, r_{1:K} | x_{1:K,0:N_k-1})$ .

Gibbs sampler samples the variables one by one by using the full conditional densities [44]. Full conditional means the probability of a variable conditioned on all other random variables and data. These full conditional distributions are mostly one dimensional, tractable and easy to sample from, depending on the model. The full conditional of a variable  $\theta$  is denoted as  $p(\theta|\cdot)$ .

In general, MCMC methods can become stuck in some local modes of the target distribution. This highly depends on the initialization of the parameter set and random variables. Even when all the initial values of the random variables are obtained from the original model where we assume the hyper-parameters are all true, there is still a chance that the sampler may stuck in a local mode of the joint distribution. A method to prevent this fact is as follows: At the beginning of iterations samples are not drawn from the full conditional  $p(r_k|\cdot)$  but from a power of it such as  $p(r_k|\cdot)^\beta$ .  $\beta$  parameter starts from a small value and increased to one as the number of iterations increase. When  $\beta$  is small then the modes of the full conditional distribution dilates and it becomes a flatter distribution therefore allows less probable samples to occur. As  $\beta$  approaches to one, samples are drawn from the exact full conditional density. This is called Simulated Tempering (ST) [44], [45]. With this method, if the initialization of the variables lead to a local mode of the joint distribution, by flattening the distribution the sampler may not stuck at that mode and hopefully starts to sample from the primary mode of the distribution and as  $\beta$  increases, it will stay in the primary mode. However even with this methodology, there is no guarantee that the sampler eventually finds the primary mode. The pseudo code of the sampler with ST is given in Figure 4.1.

In [38], Gibbs sampling is used to find the optimum alignment  $r_{1:K}^*$  with the gamma observation model  $M_1$ . The full conditional distributions  $p(\lambda_{\tau,f}|\cdot)$  and  $p(r_k|\cdot)$

```

Initialize
for  $\tau = 1$  to  $T$  do
     $\lambda_\tau \sim p(\lambda_\tau; \alpha_\lambda, \beta_\lambda)$ 
     $\{p(\lambda_\tau; \alpha_\lambda, \beta_\lambda)$  is the prior distribution for  $\lambda_\tau$  depending on the model $\}$ 
end for

Sampling from full conditionals
for  $i = 1$  to  $I$  do
    (Sampling from  $p(r_k|.)$  for  $k = 1, \dots, K$ )
    for  $k = 1$  to  $K$  do
         $r_k^{(i)} \sim p^{\beta_i}(r_k^{(i)} | x_{1:K,0:N_k-1}, \lambda_1^{(i-1)}, \lambda_2^{(i-1)}, \dots, \lambda_T^{(i-1)})$ 
    end for
    (Sampling from  $p(\lambda_\tau|.)$  for  $\tau = 1, \dots, T$ )
    for  $\tau = 1$  to  $T$  do
         $\lambda_\tau^{(i)} \sim p(\lambda_\tau^{(i)} | x_{1:K,0:N_k-1}, r_1^{(i-1)}, r_2^{(i-1)}, \dots, r_K^{(i-1)})$ 
    end for
end for

```

Figure 4.1. Gibb's Sampling with simulated tempering algorithm for multiple audio sequence alignment model.

are derived as,

$$p(\lambda_{\tau,f}|\cdot) = \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha, \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f}) \quad (4.1)$$

$$p(r_k|\cdot) = \prod_{n=0}^{N_k-1} \prod_{\tau=1}^T \mathcal{G}(x_{k,n,f}; \alpha, \frac{\alpha}{\lambda_{\tau,f}})^{[n=\tau-r_k]} \quad (4.2)$$

For this particular model, another possible ST strategy is changing the  $\alpha$  parameter during epochs. There are two meaningful reasons for using the  $\alpha$  parameter for ST. First of all, the variance of the full conditional  $p(r_k|\cdot)$  is inversely proportional with  $\alpha$  parameter. Therefore, at the beginning of the epochs, the variance of the full conditional distribution would be high which basically dilates the modes of the distribution. Secondly, from the derived formula of the full conditional distribution  $p(\lambda_{\tau,f}|\cdot)$ , it can be observed that the contribution of the data  $x_{k,n,f}$  on the conditional distribution depends on the  $\alpha$  parameter. When  $\alpha$  is small, the contribution of the data is also small and the resulting conditional distribution is closer to the prior distribution. This intuitively means that the hidden variable  $\lambda_{\tau,f}$  is not so reliable at the beginning of the epochs because the prior information is dominant in this case. Therefore starting  $\alpha$  from a small value and increasing to its original value as the number of epochs increase is also tempering.

According to the simulation results are given in Section 5.5.1, we observe that the performance of the alignment with this method depends highly on the hyperparameter choice  $(\alpha, \alpha_\lambda, \beta_\lambda)$  and in some cases the sampler becomes stuck in a local maximum hence a wrong alignment occurs. Even when enhanced with different annealing techniques, Gibbs sampler has not proven effective. Thus, we resort to a heuristic sequential search algorithm.

## 4.2. Sequential Search Algorithm

The key idea in the sequential algorithm is that we do not directly solve the difficult problem  $r_{1:K}^* = \arg \max_{r_{1:K}} \Phi(r_{1:K})$  but search  $r_{1:K}^*$  by solving some simpler problems. By 'simple', we mean to solve  $r_{1:K'}^* = \arg \max_{r_{1:K'}} \Phi(r_{1:K'})$  where  $K' < K$  (less number of sequences). By starting with  $K' = 2$  sequences and sequentially aligning one sequence at a time, we find  $r_{1:K'}^*$ , group the sequences that match with each other and freeze the relative shifts. We refer these matched sequences as a 'Cluster' and represent the  $i$ 'th cluster with  $C_i$ . The advantage of this strategy is that aligning a sequence  $k$  to a cluster  $C_i$  is no different from pairwise case. We simply compute  $\Phi(r_{C_i}, r_k)$  for all possible values of  $r_k$  and find the optimum alignment as  $r_{C_i, r_k}^* = \arg \max_{r_k} \Phi(r_{C_i}, r_k)$ .

In the sequential algorithm, we initially sort the sequences according to their lengths in decreasing order, assign the first sequence to cluster  $\mathcal{C}_1$  i.e.,  $\mathcal{C}_1 = \{1\}$ , fix the position of the first sequence i.e.,  $r_1^* = 0$  and assign all the other sequences to a list  $P = \{2, \dots, K\}$  which is the list of the sequences that are yet to be aligned.  $U$  represents the unclustered sequence list.

The algorithm starts with aligning the first sequence in the unaligned queue  $P(1) = 2$  against cluster  $\mathcal{C}_1$  hence  $\Phi(r_{\mathcal{C}_1}^*, r_{P(1)})$  is computed for all possible alignments of sequence  $P(1)$  i.e.,  $r_{\mathcal{C}_1, P(1)}^* = \arg \max_{r_{P(1)}} \Phi(r_{\mathcal{C}_1}^*, r_{P(1)})$ . Then if the alignment  $\hat{r}_{P(1)}$ , that maximizes the score function is reliable, it is also fixed as the optimum alignment  $r_{P(1)}^* = \hat{r}_{P(1)}$ , assigned to the same cluster i.e.,  $\mathcal{C}_1 = \{1, 2\}$ . In this case, the next sequence in  $P$  is aligned against the updated cluster  $\mathcal{C}_1$  instead of individual sequences i.e.,  $r_{\mathcal{C}_1, P(2)}^* = \arg \max_{r_{P(2)}} \Phi(r_{\mathcal{C}_1}^*, r_{P(2)})$ . We proceed through each unaligned sequence and the cluster is updated according to the reliability of each alignment. If any of the alignments is unreliable in the procedure, we do not assign the sequence to any cluster but to the unclustered list  $U$  and continue with the next sequence in  $P$ .

When a sequence with index  $k$  is not matched with the cluster and assigned to  $U$ , it means that it either belongs to another cluster or it does not overlap with any

of the sequences in that cluster. However, if a sequence with a higher index ( $> k$ ) is aligned after  $k$ 'th, we need to reprocess the  $k$ 'th sequence against the updated cluster to be sure whether or not it belongs to the current cluster. As a rule of thumb, after we proceed through all unaligned sequences, if the highest index in the cluster  $C_i$  is larger than the smallest index in the unclustered list  $U$  ( $\max(C_i) > \min(U)$ ), it means the cluster is updated after at least one of the sequences in  $U$ . In this case, we need to reprocess all the unclustered sequences against the updated cluster and repeat it until the smallest index in  $U$  is larger than the highest index in the cluster. When it is larger ( $\max(C_i) < \min(U)$ ), we decide that the unclustered sequences are not related to current clusters. Thus, we restart the procedure for only the unclustered sequences assigning the first sequence in the list already to a new cluster and all other sequences to the unaligned sequence list  $P$ . Then we apply these procedures until all the sequences are clustered until  $P$  is empty. The pseudocode of the sequential alignment algorithm is given in Figure 4.2.

This method is first proposed in [39] then improved in [40] with addition of clustering together with alignment. Without applying clustering, the performance of the method highly depends on the alignments of the first few sequences, thus initial ordering becomes very important. Therefore one needs to run the algorithm several times starting with different permutations to overcome this problem in the previous method. However with the described algorithm, different permutations of sequences only change the computational time but not the robustness. Therefore any initial ordering is plausible. In this work, we ordered the sequences in decreasing order according to their lengths assuming that longer sequences tend to be matched more reliably.

A demonstration of the sequential multi-sequence alignment algorithm with a set of 4 audio recordings can be found in [46].

```

Initialize:
Sort sequences,  $P = \{2, \dots, K\}$ 
 $i = 1$  (Cluster index),  $C_1 = \{1\}$ 
while  $P \neq \{\}$  do
     $U = \{\}$ 
    for  $k = P(1)$  to  $P(end)$  do
         $r_k^* = \arg \max_{r_k} \Phi(r_{C_i}^*, r_k)$ 
        if Length of overlap  $> \epsilon$  then
             $C_i = [C_i, k]$  (Add k'th seq. to the current cluster)
        else
             $U = [U, k]$  (Add k'th seq. to the unclustered list)
        end if
    end for
    if  $\max(C_i) < \min(U)$  or  $U \neq \{\}$  then
         $i = i + 1$  (Next cluster is started)
         $C_i = U(1)$  (First seq. in the unclustered list is assigned to  $C_i$ )
         $P = U(2, \dots, end)$ 
    else
         $P = U$ 
    end if
end while

```

Figure 4.2. Sequential Alignment Method.

### 4.3. Multi Resolution Audio Sequence Alignment using Sequential Monte Carlo Samplers

In this section, we introduce a SMC sampler based solution for the alignment problem that uses low resolution  $\Phi(r)$  as bridges. Here, the aim is to find the optimum alignment  $r^*$  without explicitly visiting all possible alignments. To achieve this, one needs a sampling mechanism that samples from  $\Phi(r)$  and if some of the samples would eventually hit the mode of the distribution the optimum alignment would be found.

SMC sampler is a popular sampler due to its flexibility in design and ability to sample from rough and high dimensional densities. It samples from a sequence of distributions, denoted by  $\gamma_i$ , which are called intermediate distributions [47]. At each step, the algorithm samples from the next intermediate distribution and in the last step, the resulting samples would be drawn from the target distribution which is  $\Phi(r)$  in our case. The main idea behind SMC samplers is that if the intermediate distributions in the consecutive steps are close enough to each other, they would act like a bridge and guide the samples through modes of the target density. At each step, new samples  $r_s^{(i+1)}$  are drawn from a forward Markov transition kernel  $\mathcal{K}_{i+1}(r_s^{(i+1)}, r_s^{(i)})$  where  $s$  is the sample index and  $i$  is the dimension index. Then the discrepancy between the sampling distribution and intermediate distribution is corrected using importance sampling [47]. The weight of each sample is computed as,

$$w_i(r_s^{1:i}) = w_{i-1}(r_s^{1:i-1}) \frac{B_{i-1}(r_s^i, r_s^{i-1})\gamma_i(r_s^i)}{\mathcal{K}_i(r_s^i, r_s^{i-1})\gamma_{i-1}(r_s^{i-1})}$$

where  $B_{i-1}(r_s^i, r_s^{i-1})$  is a backward Markov kernel. The increase in variance of weights indicates that some of the samples have much higher importance than others. Hence, a resampling stage is applied to get rid off the samples with small weights and replicate the ones with higher weights. A common criterion to measure this degeneracy is the effective sample size (*ESS*) which is defined in [47] as,

$$ESS = \left( \sum_{s=1}^S (w_s^{(i)})^2 \right)^{-1} \quad (4.3)$$

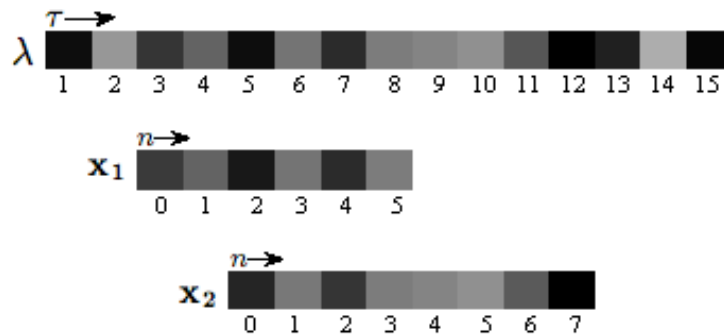
### 4.3.1. SMC Samplers for Pairwise Alignment

In this section, we extend the proposed model given in chapter 2 for multi resolution alignment and focus on pairwise cases [41]. We modify the model for multi resolution case and the matching is achieved with a SMC sampler which uses low resolution models as bridge distributions.

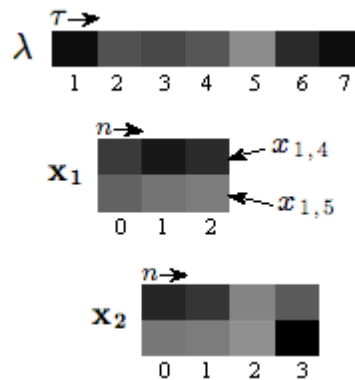
Remember that in the model, each observation coefficient  $x_{k,n,f}$  depends on only one of the hidden coefficients  $\lambda_{\tau,f}$ , if it is aligned to time  $\tau$ . To obtain lower resolution data, we modify this idea such that  $L$  number of consecutive observation coefficients depend on one hidden coefficient  $\lambda_{\tau,f}$ . To illustrate the idea, a toy example is given in Figure 4.3. In Figure 4.3 (a), there are two observed sequences that follows the original model where each observed coefficient  $x_{k,n}$  depends on only one hidden coefficient  $\lambda_{\tau}$ . On the other hand, in Figure 4.3 (b), the example is modified where  $L = 2$ . The length of each sequence is halved and as it can be observed the coefficients  $x_{1,4}, x_{1,5}, x_{2,2}$  and  $x_{2,3}$  are aligned to time  $\tau = 4$ , hence they are noisy realizations of  $\lambda_4$ . We can also interpret the second row of each sequence like a new sequence that has to be exactly aligned with the first row. From this point of view, there are 4 sequences aligned at time  $\tau = 4$ . We define  $n_L = \lfloor \frac{n}{L} \rfloor$  where  $\lfloor \cdot \rfloor$  is the floor operation and switch the local time index with  $n_L$  in the generative model which modifies the model for low resolution case. The modified template model becomes,

$$\begin{aligned} \lambda_{1:T} &\sim p(\lambda_{1:T}) \\ r_k &\sim p(r_k) = \prod_{\tau=1}^{T-N_k+1} \pi_{k,\tau}^{[r_k=\tau]} \\ x_{k,n} &\sim p(x_{k,n}|r_k, \lambda_{1:T}) = \prod_{\tau=1}^T p(x_{k,n}|r_k, \lambda_{\tau})^{[n_L=\tau-r_k]} \end{aligned} \quad (4.4)$$

It is important to mention that there are other ways to obtain low resolution



(a)



(b)

Figure 4.3. (a) Illustration of model for pairwise case with a toy example. Two observed sequences  $\mathbf{x}_1, \mathbf{x}_2$ , hidden sequence  $\lambda$  (b) Modified toy example for the modified model for half resolution in pairwise case.

sequences rather than modifying the model such as increasing window size in feature extraction procedure or downsampling before or after feature extraction process. In this work, we just modify the structure of data without changing the actual resolution.

Note that in pairwise alignment cases, by fixing one of the alignments i.e.,  $r_1$ , the search space is restricted to a one dimensional space as shown in chapter 3. Hence, score functions are one dimensional and instead of  $\Phi(r_2)$ , we will use  $\Phi(r)$  for the ease of representation.

We choose the intermediate distributions as low resolution posterior distributions denoted by  $\Phi_L(r)$  where  $L = 2^l$ ,  $l = 8, 7, \dots, 1$ . Note that the length of each  $\Phi_{L/2}(r)$  is twice the length of one step lower resolution  $\Phi_L(r)$ , i.e., length of  $\Phi_{64}(r)$  is twice the length of  $\Phi_{128}(r)$ . Hence, we need to design a forward kernel such that samples are moved from lower resolution to higher resolution. In SMC sampler framework, the choice of the forward and backward kernels are flexible so that any proposal mechanism is possible at any step of the algorithm, i.e.,  $\mathcal{K}_i(\cdot)$  do not have to be equal to  $\mathcal{K}_j(\cdot)$ .

For the forward kernel, we propose to move samples from lower resolution ( $2L$ ) to higher resolution ( $L$ ) through some smoothed distributions of  $\Phi_L$ . Defining  $Q$  as a smoothing kernel, one can obtain these distributions by applying  $Q$  several times to  $\Phi_L(\cdot)$ , i.e.,  $Q^n\Phi_L, Q^{n-1}\Phi_L, \dots, Q\Phi_L, \Phi_L$ . Illustration of the smoothed distributions through each stage and movement of a sample is shown in Figure 4.5. Note that smoothing kernel is chosen to be sparse so that one does not need to explicitly compute all values in  $Q^n\Phi_L$ , i.e., computation of a few values in  $Q^n\Phi_L$  would be enough. We applied averaging kernel for smoothing purposes and backward kernel is chosen to be equal to forward kernel in the weight update.

One issue in the design of proposal is that proposal mechanism should be different for moving samples between smooth distributions ( $Q^n\Phi_L, Q^{n-1}\Phi_L$ ) where resolution stays the same and for moving samples from low resolution ( $L$ ) to high resolution ( $L/2$ ) ( $\Phi_L, Q^n\Phi_{L/2}$ ). In the latter case, a sample in the  $(i-1)$ 'th stage in  $L$  resolution approximately corresponds to  $r_s^{(i)} \approx 2 * r_s^{(i-1)} - 1$  in the  $i$ 'th stage in  $L/2$  resolution. Hence, proposed samples at these stages are chosen around  $2 * r_s^{(i-1)} - 1$ .

Note that none of the samples represent the case  $\Phi(r = 1)$  which is the score for the sequences not overlapping. Simply by computing this value in the last step of SMC sampler where other samples are also drawn from  $\Phi(r)$  and compare with the sample of highest score, one can easily decide whether or not the sequences overlap.

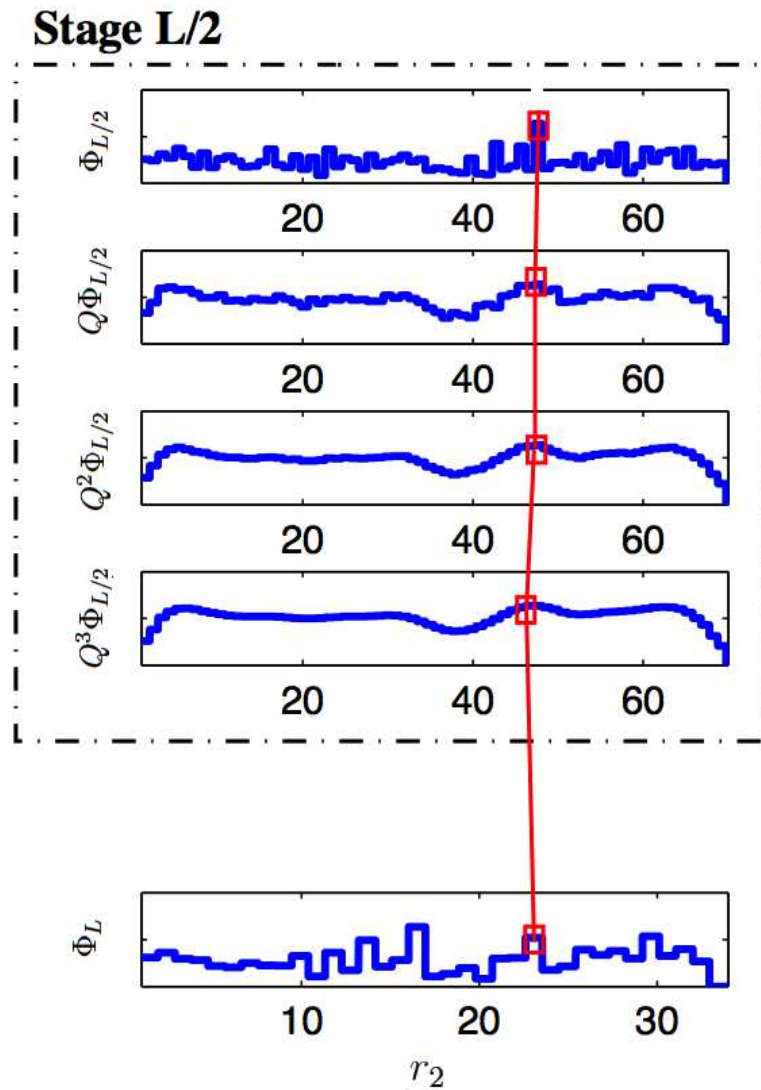


Figure 4.4. The story of a particle.

#### 4.4. SMC Samplers combined with Sequential Algorithm for Multiple Audio Sequences

We proposed multi resolution alignment using SMC samplers in Section 4.3.1 for pairwise alignment cases. Here, we extend the method for multiple alignment scenarios by using a hybrid method where we integrate SMC samplers into the sequential search algorithm introduced in Section 4.2.

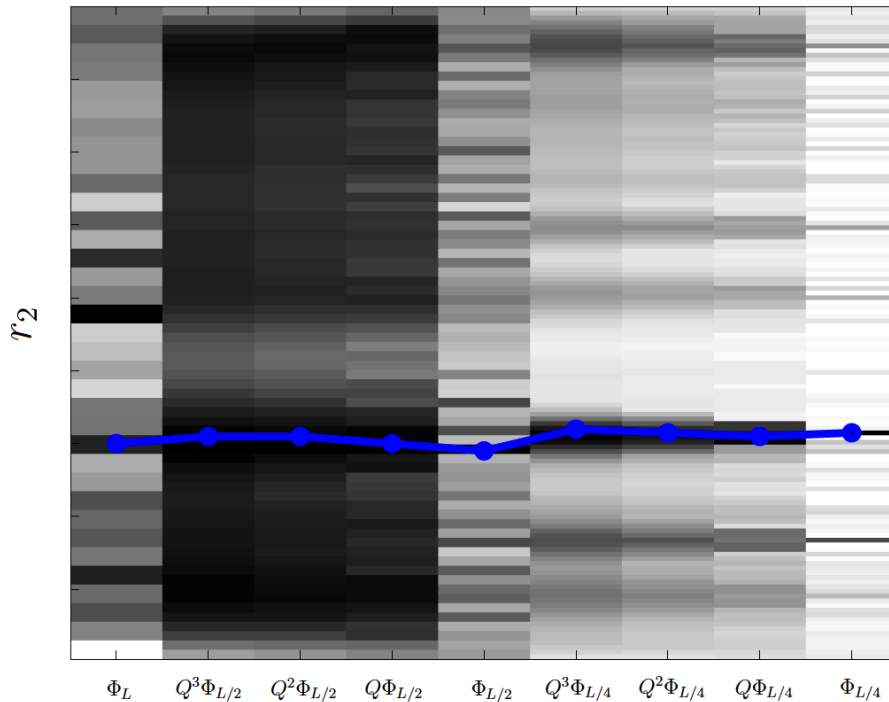


Figure 4.5. Smoothed Bridge Distribution through each stage from top view.

As explained in Section 4.2, at each step the current sequence  $\mathbf{x}_k$  is aligned against either one sequence or a cluster of sequences which leads to the following optimization problem:

$$r_k^* = \arg \max_{r_k} \Phi(r_{\mathcal{C}_i}^*, r_k) \quad (4.5)$$

Normally, the optimization in Equation 4.5 is performed by computing all possible alignments of  $r_k$  similar to a brute force and the best alignment  $r_k^*$ , is chosen as the alignment that maximizes  $\Phi(r_{\mathcal{C}_i}^*, r_k^*)$ . To integrate SMC sampler into the sequential algorithm given in Algorithm 4.2, this optimization step is replaced by the SMC sampler so that best alignment could be found without explicitly visiting all possible alignments in the  $\Phi(r_{\mathcal{C}_i}^*, r_k)$  surface. The rest of the Algorithm 4.2 stays unchanged.

An important issue regarding the performance of the SMC sampler is the initial number samples. The number of samples used in SMC sampler is determined according to the length of  $\Phi_L$  where  $L$  is the lowest resolution. For example if the length of the

sequences  $N_1 = 6500, N_2 = 7000$  and we start with a low resolution with  $L = 256$ , the length of sequences become  $\lfloor 6500/256 \rfloor = 25$  and  $\lfloor 7000/256 \rfloor = 27$  respectively where  $\lfloor \cdot \rfloor$  is the floor operation. Then the number of samples is determined as  $25+27-1=51$ . Note that the alignment of a sequence  $\mathbf{x}_i$  against a cluster  $\mathcal{C}_j$  is not different from aligning two single sequences. In both scenarios, the lowest resolution  $\Phi_L$  is computed for all possible lags using low resolution sequences as defined in Section 4.3.1.

In SMC sampler framework, intermediate distributions are usually annealed so that they become more similar [47]. Different annealing strategies are possible. Here, we anneal the intermediate distributions by adjusting the precision parameter of the corresponding model i.e.,  $w$  parameter for  $M_5$ . When precision is low, the effect of data decreases therefore sequences could be aligned with less similarity. For lower resolution models, we choose smaller precision and increase as the resolution increases. One of the major advantages of the algorithm is that, even if the corresponding alignment of the prime mode in lower resolutions is a local mode, the SMC sampler is still able to hit the prime mode in high resolution.

Note that the size of averaging kernel and/or number of appliance on the current target distributions can change over the steps of SMC sampler according to the resolution. As the resolution increases, we increase the number of appliance, hence we have more smooth intermediate distributions for higher resolution steps which is observed to enhance the performance of the algorithm.

The computation time to find an alignment estimate is an important performance criterion. We illustrate the efficiency of our model with an example using computational time for  $\Phi$  and  $\Phi_L$ . Defining the computation time for  $\Phi(r)$  for any sample  $r$  as  $T_0$ , the computation time for the  $\Phi_L(r)$  is  $T_L = \frac{1}{L}T_0$  since the length of each sequence also decreases to  $1/L$  of it. For each sample  $r_s^{i-1}$ , 2 samples are proposed for  $r_s^i$ , hence the number of required computation of smooth distributions  $Q^n\Phi_L$  is 2. Assuming there are 4 stages of the same resolution, the number of required computations is 8 between each resolution change. For  $L = 256$ , the number of increase in resolution  $\log_2 256$  is 8. Hence for one sample, the time elapsed in the end is,  $8 * (\frac{T_0}{128} + \frac{T_0}{64} + \frac{T_0}{32} + \dots + T_0)$ . The

number of samples is approximately  $1/256$  times of the original length  $N_1 + N_2$ . As a result, SMC sampler method has lower computational time compared to computing the  $\Phi(r)$  for all possible alignments, i.e.,  $(1.8125) * 8 * T_0 * (N_1 + N_2) / 256 < (N_1 + N_2) * T_0$ .

## 5. EXPERIMENTAL RESULTS

In this chapter, we evaluate the accuracy of the score functions and the proposed search algorithms i.e., Gibbs sampler, sequential multiple sequence alignment algorithm and SMC samplers method as well as deterministic similarity measures i.e., correlation and Hamming distance. In particular, we compare model-feature pairs with each other and investigate which features represent the audio data better for the alignment task in terms of immunity to noise and volume variations. Properly, we define an evaluation criterion in Section 5.1 to compare the performance of the alignments. The feature extraction procedures and the hyperparameter estimation for each model is described in Section 5.2 and Section 5.3 respectively.

In the first part of the experiments in Section 5.4, we test each model-feature pair with pairwise real data sets. The performances of cross-correlation and Hamming distance measures for these data sets are also given in that section. Then in the second part of the experiments, we apply the proposed search algorithms for data sets that contain multiple sequences. This part is divided into two subparts:

- Experiments with artificially produced data sets (Section 5.5.2)

The sequences are artificially formed from music or speech files and adding a structured noise. The experiments are conducted for low SNR and high SNR cases.

- Experiments with real data sets (Section 5.5.3)

Real data sets contain multiple sequences with multiple shots taken by each recorder.

Beside the sequential search algorithm and SMC sampler methods, we also give a comparison between these methods and a correlation based baseline method. It is important to mention that in this work, we collaborated with Singular Software Company that has a commercial application for video synchronization and the real data sets (both pairwise and multiple) are provided by them due to our collaboration.

### 5.1. Evaluation Criterion

For evaluation of the alignment performance, we first define a correct alignment between two sequences  $i$  and  $j$ . The correct alignment occurs in one of the two cases: In the first case, the sequences do not match in the ground-truth and the estimated alignments  $\hat{r}_i$  and  $\hat{r}_j$  result in a non-overlapping alignment, in the second case the sequences match in the ground-truth and the relative distance between the estimated alignments  $\hat{r}_i$  and  $\hat{r}_j$  is equal to distance in the ground-truth, i.e.,  $r_i - r_j = \hat{r}_i - \hat{r}_j$ . Note that sometimes the estimated relative distance misses the ground-truth value with a small error  $\zeta$ , i.e.,  $|r_i - r_j| - |\hat{r}_i - \hat{r}_j| < \zeta$ . In experiments, we choose  $\zeta = 2$  frames and such cases are assumed to be correct.

We define an indicator function  $\phi(\hat{r}_i, \hat{r}_j)$  in (5.1) that determines if two sequences  $i$  and  $j$  are mutually correctly aligned.

$$\phi(\hat{r}_i, \hat{r}_j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are correctly aligned} \\ 0, & \text{otherwise} \end{cases} \quad (5.1)$$

The alignment performance criterion  $\Omega(\hat{r}_{1:K})$ , the total alignment score, is then defined in (5.2) as the number of true pairwise alignments over total number of pairs.

$$\Omega(\hat{r}_{1:K}) = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \phi(\hat{r}_i, \hat{r}_j) \quad (5.2)$$

Highest score to be achieved is "1" where all the sources are aligned perfectly and lowest score is "0" where no sources are aligned correctly.

As mentioned in Section 3.4, as the precision of the observation model decreases, the unreliable region at the sides of  $\Phi(\mathbf{r})$  functions becomes wider. In experiments, we assume reliable alignments when the overlap between sequences is larger than 3

seconds for low SNR cases and larger than 2 seconds for high SNR cases.

## 5.2. Selected Features and Feature Extraction Procedure

In this work, we applied STFT (40ms windows with 20ms hop-size) to the audio signals then further apply various methodologies to obtain different feature representations. There are 8 feature sets that are used in our simulations that can be listed as;

- *Feature 1*: Energy in sub-bands (Positive Coefficients)

The STFT is divided into 6 logarithmically spaced sub-bands and the energy in each sub-band of the magnitude spectrum is computed by summing the squared values of coefficients in one band. We denote this feature with " $F_1$ ".

- *Feature 2*: Positive Spectral Difference (Positive Coefficients)

First difference of the STFT is obtained through time and the positive values are summed through frequency [33]. This feature is particularly used in onset detection. We denote this feature with " $F_2$ ".

- *Feature 3*: MFCC (Real Coefficients)

In the MFCC extraction procedure, the STFT coefficients are first transformed into mel scale, then the discrete Cosine transform (DCT) is applied to the logarithm of the energy of the coefficients. We denote this feature with " $F_3$ ".

- *Feature 4*: First difference of the energy in subbands both through frequency and time (Real Coefficients)

The STFT is divided into 17 or 33 logarithmically spaced subbands, then the first difference of the energies is obtained through frequency and then in time. We denote this feature with " $F_4$ ". The procedure is shown in Figure 5.1.

- *Feature 5*: Thresholding the first difference of the energy in subbands both through frequency and time (Binary Coefficients) [15]

The resulting coefficients of  $F_4$  are thresholded [15]. We denote this feature with " $F_5$ ". The procedure is shown in Figure 5.2.

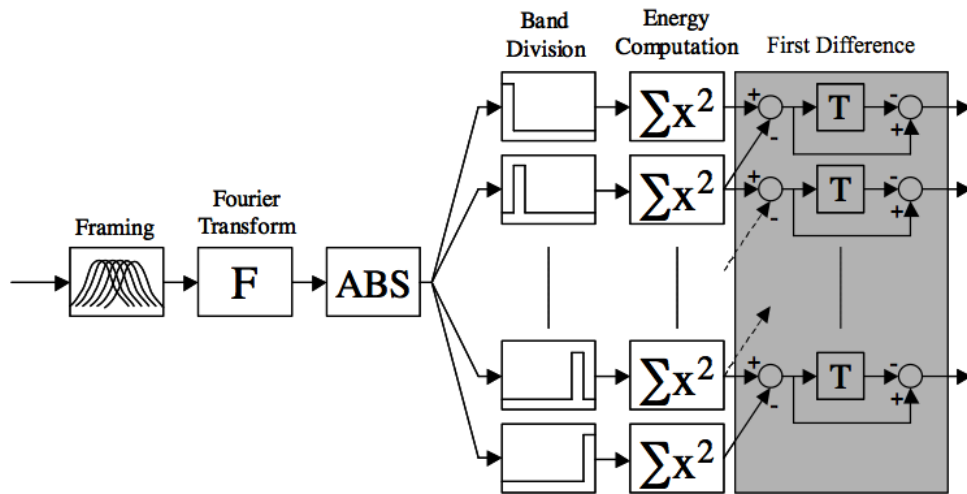


Figure 5.1. Extraction of feature  $F_4$  - Real Features.

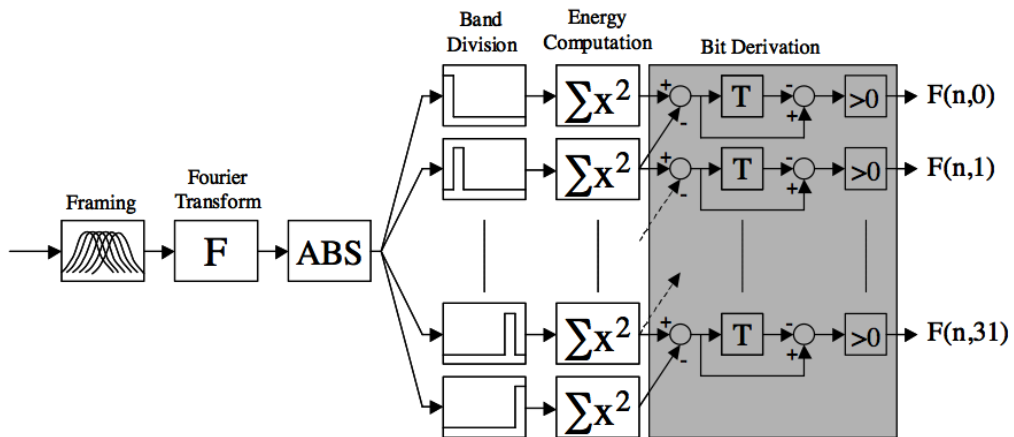


Figure 5.2. Extraction of feature  $F_5$  - Binary Features.

- *Feature 6*: First difference of the energy in subbands of the spectrum through time (Real Coefficients)

The STFT is divided into 14 logarithmically spaced sub-bands and the first difference of the energies is obtained through time. We denote this feature

with " $F_6$ ".

- *Feature 7*: The thresholded energy in subbands (Binary Coefficients)

The STFT is divided into 14 logarithmically spaced subbands and the energy in the subbands are thresholded. We denote this feature with " $F_7$ ".

- *Feature 8*: Quantized coefficient energy in each subband (Categorical Coefficients)

The STFT is divided into 6 logarithmically spaced subbands, then the energy of each subband is quantized with  $\mu$ -law into  $Q=6$  levels. The  $x_{1:Q,n,k}$  is a vector for which only one element of the vector is active and the rest of the elements are equal to zero. As an example, if there are  $Q = 3$  levels and the second level is selected, the vector is,  $x_{1:Q,n,k} = \{0, 1, 0\}$ . We denote this feature with " $F_8$ ".

In Figure 5.3, three aligned overlapping sources and their respective feature sequences  $F_2$  and  $F_3$  are shown as an example.

### 5.3. Hyperparameter Choice

Model based alignment problem can be viewed as an optimization problem not only on the possible alignments  $r_{1:K}$  but also on the hyperparameters  $\Theta$ . So far, we fixed the hyperparameters specific to the data sets by trial and error methods, and optimization is applied on the possible alignments. By this way, the problem becomes tractable even for multiple alignment cases because the alignments  $r_{1:K}$  are discrete. However actual problem is finding the optimum alignments blindly, without any knowledge on the hyperparameter sets.

For the estimation of hyperparameters, we used an iterative Newton's method on the score functions in [39] to obtain optimum hyperparameter sets for each model where the ground truth is known. However, the surface over parameters and alignments is pretty rough therefore optimization procedure highly depends on initial values and fail most of the time. In addition, for real data sets where the ground truth is usually not known, this method is not applicable.

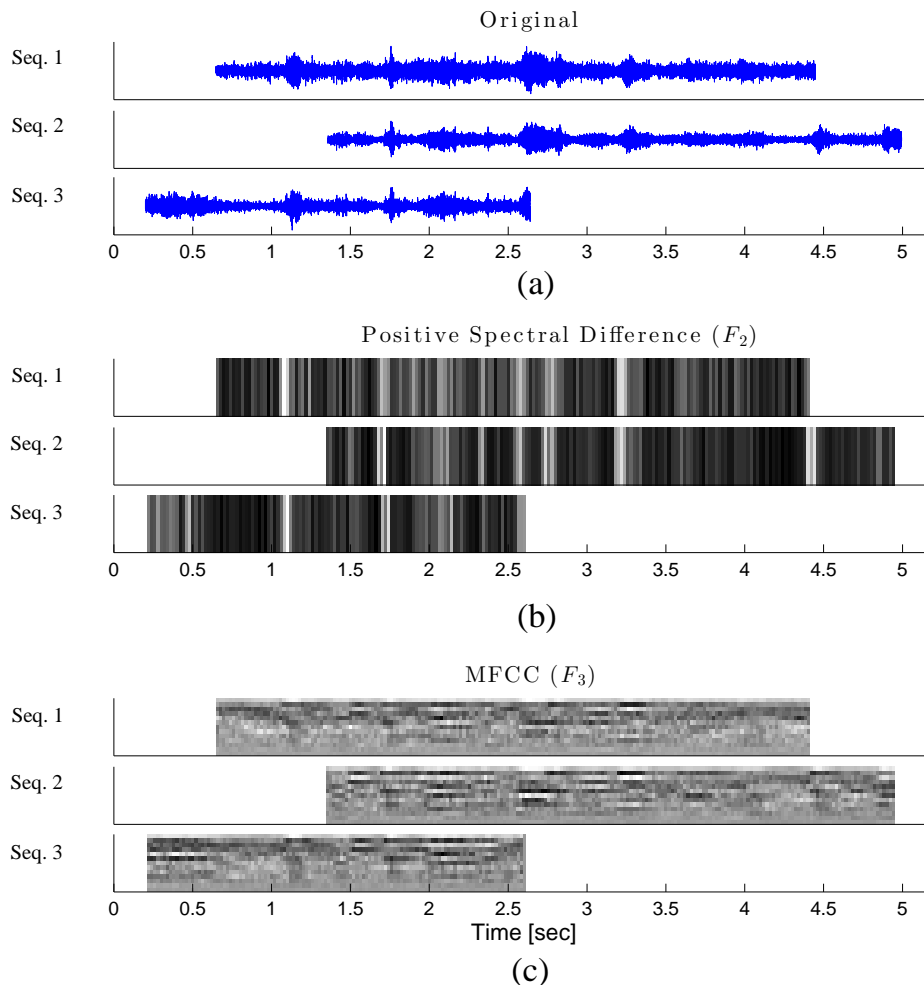


Figure 5.3. Sources and features illustration: (a) Three aligned time domain sequences (b) Extracted positive spectral difference ( $F_2$ ) for aligned sequences (c) Extracted MFCC coefficients ( $F_3$ ) for aligned sequences.

Another approach for finding optimum hyperparameters and alignments at the same time is a line search strategy [39]. We set an interval and a grid for each hyperparameter and find the optimum alignment for each possible hyperparameter combinations. The highest likelihood score possibly occurs when we hit the optimum hyperparameter settings and the alignment if we are searching the right interval with a fine step size. That is actually true if our model is correct and describe the data set well enough. An example of this approach is given for model  $M_5$  in Figure 5.4. In this example, there are two overlapping sequences and  $\Phi_{M_5}(r)$  is computed and plotted

where the  $w$  parameter is changed linearly from 0.6 and 0.8. It can be observed that the maximum of all likelihoods occur at  $w = 0.64$ , and it gives the true alignment result. However, for values  $w > 0.7$ , the alignment can not be found for this data set. Unfortunately, this approach is feasible only for the pairwise alignment cases (given that the model describes the data very good). However, this method becomes intractable for the multiple alignment.

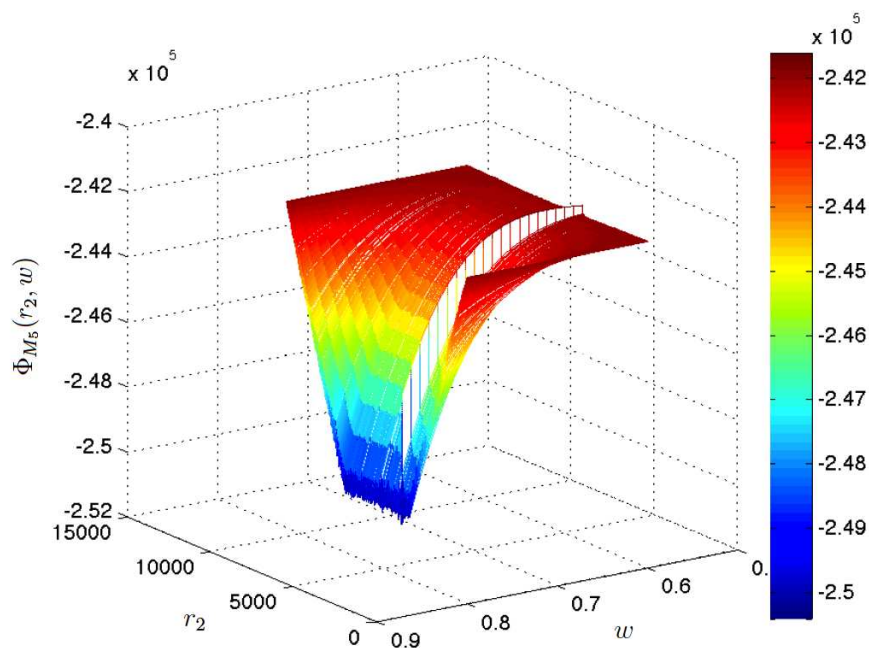


Figure 5.4. 3D plot of  $\Phi_{M_5}(r_2, w)$  over  $r_2$  and  $w$ .

Note that the performance of the algorithm is mostly affected by the false positive and false negative alignments and so there is a trade off in the choice of hyperparameter sets. A false positive alignment occurs when two sequences are not overlapping at the ground truth but they are estimated as aligned and overlapping with each other. Conversely a false negative alignment occurs when two sequences are not matched at the ground truth but they are estimated as aligned and overlapping with each other. When the observation variance is too high with the choice of hyperparameters of the related model, two non-overlapping sequences might match with each other which causes false positive results. In the opposite case, where the observation variance is

too small, two overlapping sequences might not be aligned especially if they are not so similar i.e., low SNR, hence causing false negative alignments. In this work, we defined two parameter sets for each model, one for low SNR cases ( $-4\text{dB} < \text{SNR} < -2\text{dB}$ ) and one for high SNR cases ( $9\text{dB} < \text{SNR} < 11\text{dB}$ ). We tuned these parameters using pairwise real audio data sets with various types and amounts of noise. The hyperparameters used for each model are listed in table 5.1 for low and high SNR cases.

Table 5.1. Hyperparameter choices for each model for low SNR and high SNR cases.

Model-Feature	Low SNR case				High SNR case			
	$\alpha_\lambda$	$\beta_\lambda$	$\alpha$	$w$	$\alpha_\lambda$	$\beta_\lambda$	$\alpha$	$w$
$M_1-F_1$	2.5	50	30	-	2.5	1	40	-
$M_1-F_2$	2.5	50	30	-	2.5	70	35	-
$M_2-F_3$	2.5	40	-	-	2.5	80	-	-
$M_2-F_4$	8	80	-	-	8	80	-	-
$M_2-F_6$	4	80	-	-	2	85	-	-
$M_3-F_3$	-	450	0.03	-	-	450	0.06	-
$M_3-F_4$	-	35	0.004	-	-	35	0.01	-
$M_3-F_6$	-	45	0.004	-	-	45	0.02	-
$M_4-F_5$	0.5	0.5	-	-	2	2	-	-
$M_4-F_7$	0.8	0.2	-	-	2	0.5	-	-
$M_5-F_5$	0.5	-	-	0.62	0.5	-	-	0.75
$M_5-F_7$	0.23	-	-	0.62	0.2	-	-	0.75

#### 5.4. Pairwise Audio Sequence Alignment Results

In this section, the experimental results are given for the pairwise real data sets. There are 20 data sets that include sequences with various types of degradations such as additive wind noise or spectral distortion due to equalizer settings. The ground truth alignments are obtained by careful listening.

As mentioned in Chapter 2, the observation model directly determines the type of the feature set to be used i.e., real, positive, binary or categorical. Therefore some

of the feature sets are appropriate for more than one model such as *Feature 6* (*model 4* and *model 6*). Experiments are conducted with different model-feature pairs with the real pairwise data sets and the performances are listed in the Table 5.2 following the hyperparameter settings given in Table 5.1. The model-feature pairs are named in the following way; *Model a* and *Feature b* pair is represented with  $M_a F_b$  i.e., *Model 2-Feature 3* pair  $\rightarrow M_2 F_3$ .

Note that in our initial experiments the real data sets, we tried several different hyperparameter settings for each model and eventually the model-feature pairs are able to find the true alignment for each pairwise real data set. Then we investigated the possibility to determine one set of parameters for each model-feature pair to handle all alignment scenarios (low SNR, high SNR, overlapping, non-overlapping) for practical purposes unfortunately the results were not encouraging i.e., no single parameter set is enough to solve all alignment scenarios. This is actually not incomprehensible because the parameter settings for a noisy set of sequences should lead to a low precision for a model whereas for a clean set of sequences the parameter settings lead to a high precision (see Section 2.1). Therefore we defined two different settings for each model for varying noise conditions in Table 5.1.

To compare the performances of model-feature pairs with the well known similarity measures correlation and Hamming distance, we also compute generalized cross-correlation and Hamming distance for each data set. Note that generalised cross-correlation is applied in the feature domain for  $F_1, F_2, F_3, F_4, F_5, F_6$  and  $F_7$  features, whereas Hamming distance is computed with  $F_6$  and  $F_7$  feature sets. The results are given in Table 5.3.

The results in Table 5.2 and Table 5.3 show that correlation and Hamming distance measures have high performances, however with the correct choice hyperparameters, some of the model-feature pairs have higher performances than correlation and Hamming distance methods. Note that all the pairwise data sets consist of overlapping sequences hence we are able compare the performances of model-feature pairs and deterministic similarity measures. An interesting observation on the alignment

Table 5.2. Alignment performances of each model-feature pair for 20 real data sets.

Model-Feature Pair	Number of correct alignments
$M_1F_1$	17
$M_1F_2$	18
$M_2F_3$	13
$M_2F_4$	13
$M_2F_6$	11
$M_3F_3$	17
$M_3F_4$	18
$M_3F_6$	14
$M_4F_5$	15
$M_4F_7$	13
$M_5F_5$	19
$M_5F_7$	12
$M_{5,1}F_5$	19
$M_6F_8$	8

performances of model-feature pairs is that whenever a false alignment occurs, the result is chosen as the non-overlapping alignment of the sequences (false negative) rather than aligned to a wrong position. That is obviously not the case for the deterministic similarity measures. Among the pairwise real data sets, there is one particular data set which leads to one of the difficult scenarios. In this data set, the sequences are highly exposed to the wind noise. We observe some of the model-feature pairs, most of the correlation based (except for correlation with  $F_2$ ) and Hamming distance with  $F_6$  alignments fail for this data set.

An important performance issue for an alignment algorithm is the number of false positives or negatives in the alignment estimates. The model based approach (model-feature pairs) is tested against false positive/negative analysis on a subset of the Data

Table 5.3. Alignment performances using cross correlation and Hamming distance among 20 real data sets.

Similarity Measure	Number of correct alignments
Correlation with $F_1$	13
Correlation with $F_2$	18
Correlation with $F_3$	16
Correlation with $F_4$	17
Correlation with $F_5$	17
Correlation with $F_6$	16
Correlation with $F_7$	11
Hamming with $F_5$	18
Hamming with $F_7$	12

set 2 given in Section 5.5.3. This subset contains 20 recordings (6 overlapping pairs) recorded by two recorders (10 recordings each). Model-feature pair  $M_5F_5$  is applied to each possible combination of 2 sequences with  $w = 0.7$  and the results are listed in Table 5.4. The alignment results are quite encouraging.  $M_5F_5$  is able to detect whether the sequences overlap or not. In addition, if the sequences are overlapping, the true alignments are estimated. There are two exceptional cases in the data set where the model fails to find true alignments;

- When one or both sequences are short ( $< 2$  seconds)
- Repeating speech of the same person in different records

As number of overlapping parts of sequences increase, the estimation becomes more robust (see Section 3.4). When the sequences are shorter than 2 seconds, the alignment estimation highly depends on the hyperparameter choice of the particular model. Since the length of the data is not sufficient, the effect of data sets on alignment is lower. We observe false positives in such cases. As an example, Rec8 from Recorder1 is 1.7 seconds long and the alignment result suggests that this sequence is overlapping both with Rec4 and Rec7 from Recorder2 although these sequences do not overlap.

Table 5.4. False Positive/Negative Analysis. *O*: Overlap at the true alignment, *NO*: Not overlapping, ***FP***: False positive, ***FN***: False negative.

<b>Recorder 1</b>	<b>Recorder 2</b>									
	Rec1	Rec2	Rec3	Rec4	Rec5	Rec6	Rec7	Rec8	Rec9	Rec10
Rec1	<i>NO</i>	<i>O</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec2	<i>NO</i>	<i>NO</i>	<i>O</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec3	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec4	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>O</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<b><i>FP</i></b>	<i>NO</i>
Rec5	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec6	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>O</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec7	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>O</i>	<i>NO</i>	<i>NO</i>
Rec8	<i>NO</i>	<i>NO</i>	<i>NO</i>	<b><i>FP</i></b>	<i>NO</i>	<i>NO</i>	<b><i>FP</i></b>	<i>NO</i>	<i>NO</i>	<i>NO</i>
Rec9	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>O</i>	<i>NO</i>
Rec10	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>	<i>NO</i>

There is one very interesting pair of recordings in the data set (Rec4 from Recorder1 and Rec9 from Recorder2) for which the  $M_5F_5$  model-feature pair failed. For this pair of sequences, algorithm suggest an overlapping alignment result although they do not (false positive). These sequences are also long enough to not to suffer from the previously explained case. In both recordings, a director counts down and an actress repeats her lines with nearly same rhythmic structure and tone. In Rec4 from Recorder1, the director is a male however in Rec9 from Recorder2, the director is a female. But apart from this, the parts that contain the voice of the actress are nearly identical i.e., it is not possible to distinguish sequences by careful listening. By fine tuning the hyperparameter  $w$  of the model  $M_5$ , non-overlapping alignment (true estimation) results can be obtained. However there is a trade-off in tuning the parameters as mentioned in Section 3.4, it should find matchings in noisy cases so the precision should be low enough, but also it should be high enough to prevent such wrong alignments. In the rest of the experiments, we follow the values given in the Table 5.1 for the hyperparameter choices.

## 5.5. Multiple Audio Sequence Alignment Results

In this section, the experimental results are given for multiple audio sequence alignment scenarios. As we mentioned, our initial efforts on optimization of the alignments of multiple sequences involve Gibbs sampling method and in the first part (Section 5.5.1) the simulation results are given for both synthetic and real data. Then in the second part (Section 5.5.2), we exhaustively test and compare the robustness of model-feature pairs and SMC sampler for multiple audio sequences method that is proposed in Section 4.4 under various scenarios with artificially produced data. In the last part (Section 5.5.3), the best model-feature pair and SMC sampler are tested with two large real data sets to illustrate their performance under real life scenarios.

To compare the performances of model-feature pairs and SMC sampler with a baseline method, we integrate correlation into our sequential algorithm. To be precise, instead of simple correlation, we prefer the absolute value of the correlation coefficient as the similarity measure since it gives normalized scores between 0 and 1 for each lag. The sequences are merged together if there exists a match by simply taking the mean of the aligned coefficients of sequences [29]. For the feature extraction part, the positive spectral difference  $F_2$  is computed due to its known robustness in musical signal processing applications [33]. We manually set the threshold to discriminate overlapping and non-overlapping sequences. For this purpose, we tuned the threshold parameter according to the data set until we obtain the highest performance. The baseline method is applied on both artificial and real data sets in Section 5.5.2 and Section 5.5.3, respectively, and the alignment performances between baseline method, model-feature pairs and SMC sampler method is given.

Note that comparison with baseline Hamming distance method by integrating into sequential method is not trivial. Such simple similarity measures are able to align only two sequences which requires a merging step for the aligned sequences. However Hamming distance is defined for binary observations hence merging aligned sequences is not possible. As a result, we only compare our methods with baseline correlation method.

### 5.5.1. Results for Gibbs Sampler with Simulated Tempering

Simulation results are obtained on both synthetic and artificial data sets. The synthetic data is generated from the model  $M_1$  with the hyperparameter set  $\{\alpha_\lambda = 1.2, \beta_\lambda = 10, \alpha = 14\}$ . Several experiments are conducted in this setup with various different settings of hyperparameters. It is observed that Gibbs sampler, without any tempering or annealing, often gets stuck in a local maximum of the posterior distribution  $p(\mathbf{r}, \lambda | \mathbf{x})$ . Simulated tempering approach for Gibbs sampler leads to better performance depending on the tempering strategy. There is no guarantee that the sampler eventually samples from the prime mode of the posterior. One way to deal with this fact is to run Gibbs sampler several times from different initial points and accept the most probable outcome using  $\Phi_{M_5}(\mathbf{r})$ .

The model successfully matches signals even when some samples are missing. Figure 5.5 illustrates such a scenario for a pair of synthetic sequence. Here some of the samples of clip 2 are deleted from the part that overlaps with clip 1. The clips are shown in Figure 5.5.a. The probability of  $r_2$  for each possible alignment given the  $r_1$  is shown in Figure 5.5.b. Aforementioned, simulated tempering flattens the full conditional density  $p(r_k | \cdot)$  and sampling from tempered density results in a rather uniform samples. However, as number of epochs increases the samples are drawn from the original full conditional, therefore sampler mostly samples from the mode of the distribution. This fact is illustrated in Figure 5.5.c that shows  $r_2$  estimate at each epoch. Figure 5.5.d shows the resulting alignment of the clips.

For real data simulations, a data set with  $K = 3$  sequences is used. The sequences are recorded in a concert by three microphones during one song and the SNR is higher than 10dB.  $F_1$  feature is used to represent audio files with only three subbands. The parameter setting is chosen as  $\alpha_\lambda = 3$ ,  $\beta_\lambda = 1.5$  and  $\alpha = 5$ . Gibbs Sampler has 1000 epochs and both of the simulated tempering methodologies are applied. The Gibbs sampling procedure is started with different initial  $r_{1:K}$  estimates for 20 times. Then the score of the particular alignment with  $\Phi_{M_5}(r_{1:K})$  is computed. The alignment with the maximum score is chosen as the optimum alignment result. In this simulation, the

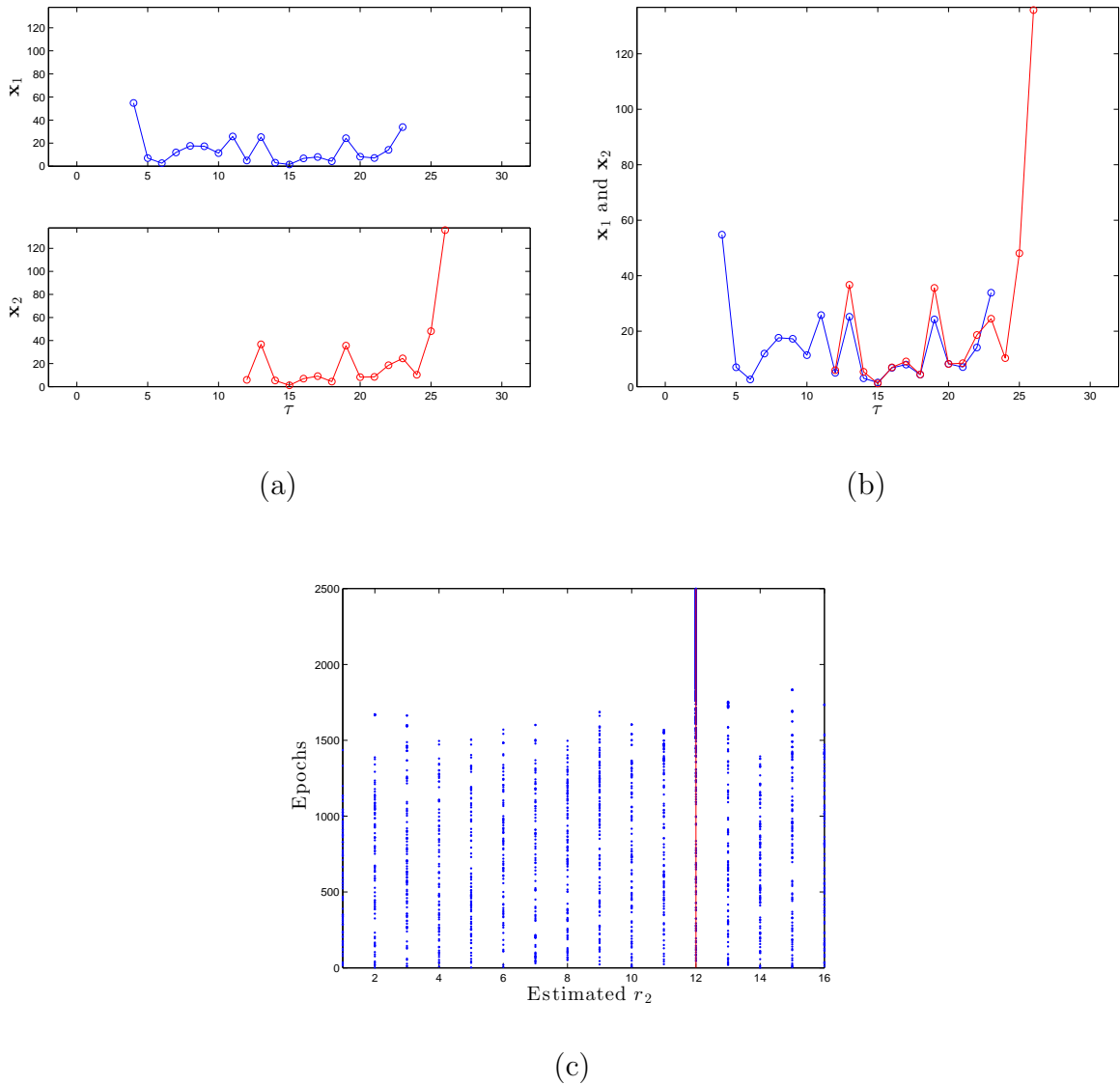


Figure 5.5. Alignment on synthetic data. (a) Generated Sequences  $\mathbf{x}_1$  and  $\mathbf{x}_2$  (b) Gibbs sampler (SA) alignment estimation with 2500 epochs(c)  $r_2$  estimate at each epoch.

alignment parameters for each source are estimated as  $r_1 = 50$ ,  $r_2 = 121$  and  $r_3 = 170$ . Alignment results are shown in Figure 5.6.a with time domain signals and in Figure 5.6.b with features.

Note that in this high SNR scenario, the alignment results are successful. However, the Gibbs sampler, even with simulated tempering approaches, often sticks at a

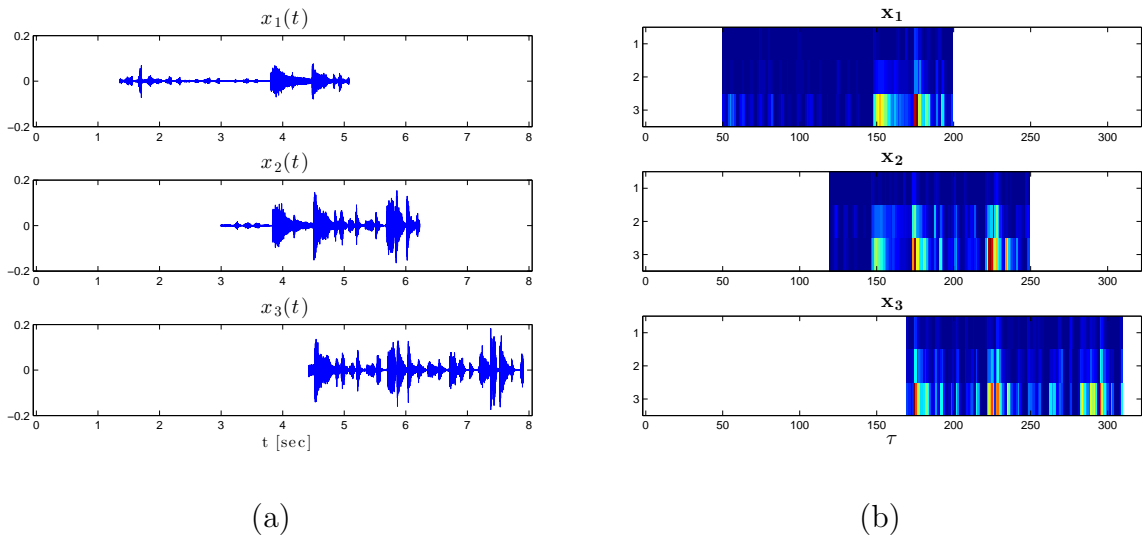


Figure 5.6. Alignment on real data using Gibbs: 1000 Epochs, 20 times. (a) Aligned time domain signals  $x_1(t)$ ,  $x_2(t)$  and  $x_3(t)$  (b) Aligned features  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$ .

local maximum of the posterior distribution  $p(\mathbf{r}, \lambda | \mathbf{x})$ . This is due to the fact that the optimization surface is very rough. To illustrate this issue, the  $\Phi_{M_1}(\mathbf{r})$  score function is computed and plotted for all possible alignments in Figure 5.7. Since the relative shifts (alignments) are searched, the score function is computed for all relative shifts of  $r_2$  and  $r_3$ .

### 5.5.2. Experiment 1: Artificially Produced Data

The aim of this experiment is to compare the robustness of model-feature pairs under different noise and volume conditions and to compare the performances of sequential search algorithm, SMC sampler method and the correlation based method. In each experiment,  $K$  sequences with random starting points and lengths are artificially produced from one of the four source files; 3 music files from rock, jazz and classical genres and a speech file. By using different types of audio files, we are also able to analyze the effect of the structure of the audio file to the performance of the alignment. For example, percussive sounds usually produce high peaks in the STFT therefore easier to align. Rock music songs usually contain much more percussive or drum sounds than classical music hence it is expected that the performance for the classical music

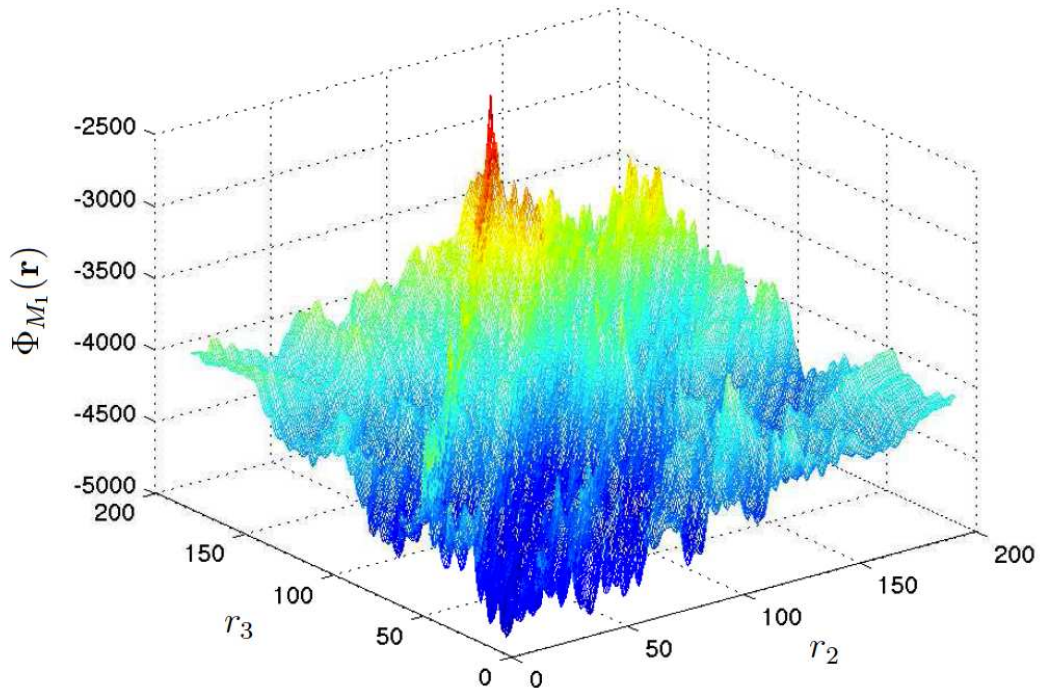
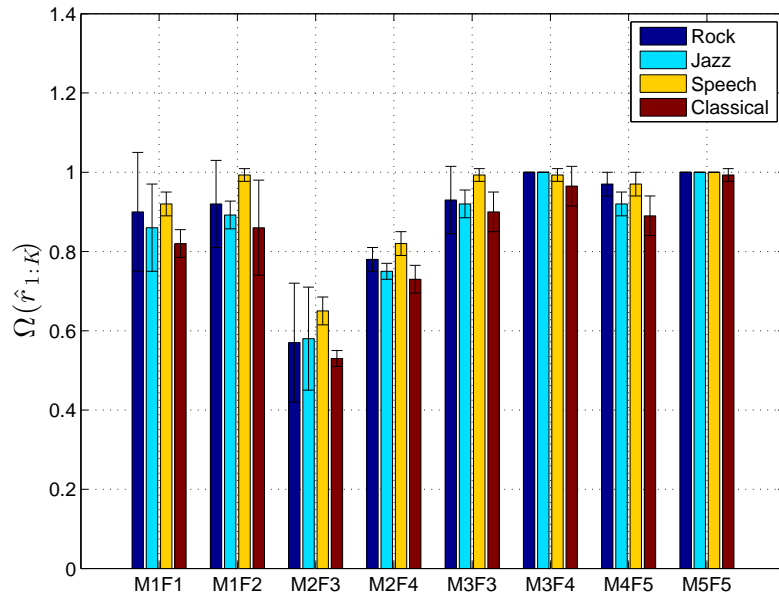


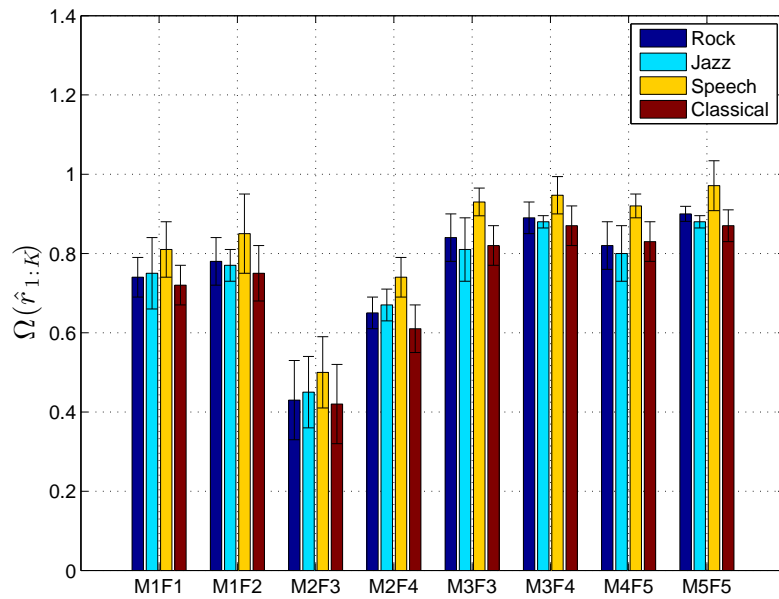
Figure 5.7. The Scoring Function  $\Phi_{M_1}(\mathbf{r})$  for all possible values of alignments.

experiments to be worse. One other issue is that music files usually have repetitions i.e., chorus parts, which might lead to misalignments between sequences whereas speech files do not have such repetitions that prevents this type of error.

The sequences are artificially formed in the following way:  $K = 8$  sequences are formed in 10-40 seconds range at each experiment, from a stereo file of length 2 minutes with 8kHz sampling rate. Equal number of sequences are formed from the right and left channels (4 sequences from each channel). The starting points ( $r_{1:K}$ ) and the lengths of the sources ( $N_{1:K}$ ) are chosen randomly at each experiment. Each sequence is multiplied with a random volume variable  $m_k^1$  which is in the range  $0.5 < m_k^1 < 1$ . To simulate noisy conditions, we used bar ambiance recordings as a structured noise. They are divided into clips following the same lengths and added to each sequence. Noise sources are also multiplied with a random volume variable  $m_k^2$  which is set randomly in 2 different ranges to simulate different SNR cases i.e, for high SNR cases;  $9\text{dB} < \text{SNR} < 11\text{dB}$ , for low SNR cases;  $-4\text{dB} < \text{SNR} < -2\text{dB}$ .



(a)



(b)

Figure 5.8. Performances of different model-feature pairs in mean alignment scores and standard deviations for rock, jazz, classical music and speech a) for high SNR cases ( $\approx 10\text{dB}$ ) b) for low SNR cases ( $\approx -3\text{dB}$ ).

For each model-feature pair, the sequential alignment algorithm is applied and the total alignment scores ( $\Omega(\hat{r}_{1:K})$ ) are computed. Aforementioned, we defined two hyperparameter sets for each model, one for low SNR cases and one for high SNR cases which are given in Table 5.1. The experiments are conducted 20 times for both cases. The mean of the alignment scores ( $\Omega(\hat{r}_{1:K})$ ) and associated standard deviations for each source file are plotted for high and low SNR cases in Figure 5.8(a) and Figure 5.8(b) respectively. Note that multinomial observation model  $M_6$  is excluded in the computations due to its unsatisfactory performance in the pairwise alignment experiments. Accordingly, we remove the categorical feature  $F_8$ . The  $F_6$  and  $F_7$  features are also excluded in the experiments since the counterparts for these features have higher performances i.e.,  $F_4$  has a better performance than  $F_6$  and  $F_5$  has a better performance than  $F_7$ .

The results suggest that the sequential algorithm is able to find the true alignments in most of the experiments even in low SNR cases. For a specific model-feature pair, there is not much difference in the performance for different audio styles i.e., rock, jazz, classical music, speech. In general, the performance is worst for classical music as expected. We observe that the repetitions in music i.e., chorus parts, cause some false positives or misalignments especially in low SNR cases. Aforementioned, such errors do not occur in speech experiments hence performances for speech are higher.

When we compare the models  $M_2$  and  $M_3$ , we see that  $M_3$  has significantly higher results for both  $F_3$  and  $F_4$  features. It is also the case between models  $M_4$  and  $M_5$  where  $M_5$  has higher performance. The common trait between  $M_2$  and  $M_4$  is that there is no explicit precision parameter to control how much observations deviate from hidden sequence. Hence, optimizing the hyperparameters for different SNR cases is rather difficult for these models.

Among the model-feature set pairs, except  $M_2F_3$  and  $M_2F_4$ , all the model-feature pairs have similar performances. We can observe that  $M_5F_5$  has a slightly better performance than others in both SNR cases. To evaluate the statistical significance of  $M_5F_5$  performance, we applied a hypothesis testing on the mean alignment scores. For

this case, we defined the null hypothesis,  $H_0$  as; each model-feature pair is better than  $M_5F_5$ . Then we applied a one sided t-test by comparing the results of  $M_5F_5$  to all other model-feature pairs for each audio source and the p-values are listed in Table 5.5 and Table 5.6 for low SNR and high SNR cases respectively. It can be observed that in low SNR cases, with %95 significance level,  $M_5F_5$  method is significantly better than all methods except  $M_3F_4$  in all audio sources. Only in classical music  $M_5F_5$  is significantly better than  $M_3F_4$  with %99 significance level. The results for high SNR cases also show similar significance for model-feature pair  $M_5F_5$ . Hence among the model-feature set pairs, we decide that  $M_5F_5$  is significantly better than other model-feature pairs.

Table 5.5. p-values for comparison of  $M_5F_5$  performance with all other model-feature pair performances in low SNR case.

Model-Feature	Rock	Jazz	Classical	Speech
$M_1F_1$	0.0	0.0014	0.0	0.0
$M_1F_2$	0.0	0.0028	0.0	0.0
$M_2F_3$	0.0	0.0	0.0	0.0
$M_2F_4$	0.0	0.0	0.0	0.0
$M_3F_3$	0.0136	0.0417	0.0082	0.0006
$M_3F_4$	0.3395	0.5	0.0904	0.5
$M_4F_5$	0.0021	0.0225	0.0015	0.0041

Table 5.6. p-values for comparison of  $M_5F_5$  performance with all other model-feature pair performances in high SNR case.

Model-Feature	Rock	Jazz	Classical	Speech
$M_1F_1$	0.0	0.0	0.0	0.0
$M_1F_2$	0.0021	0.0	0.0326	0.0007
$M_2F_3$	0.0	0.0	0.0	0.0
$M_2F_4$	0.0	0.0	0.0	0.0
$M_3F_3$	0.0008	0.0	0.0326	0.0001
$M_3F_4$	0.5	0.5	0.0326	0.5
$M_4F_5$	0.0001	0.0	0.0001	0.0

Having decided that  $M_5F_5$  has significantly higher performance than other model-feature pairs, we implemented the SMC sampler search algorithm with  $\Phi_{M_5}$  score function and apply with feature  $F_5$ .  $M_5F_5$  and accordingly  $\Phi_{M_5}(\mathbf{r})$  has two additional advantages;

- Feature  $F_5$  is immune to volume changes [15].
- Model  $M_5$  has two hyperparameters<sup>10</sup>  $\alpha_\lambda$  and  $w$ . By assuming 1's and 0's are equiprobable in the latent variables,  $\alpha_\lambda$  can be chosen as 0.5 which leads to a single parameter  $w$  for the model.

We compare the performances of the  $M_5F_5$  model-feature pair, SMC sampler and the baseline correlation method with the same experimental setup and the alignment performances are shown in Figure 5.9. It is observed that both SMC sampler and  $M_5F_5$  have better performances than the baseline correlation method.

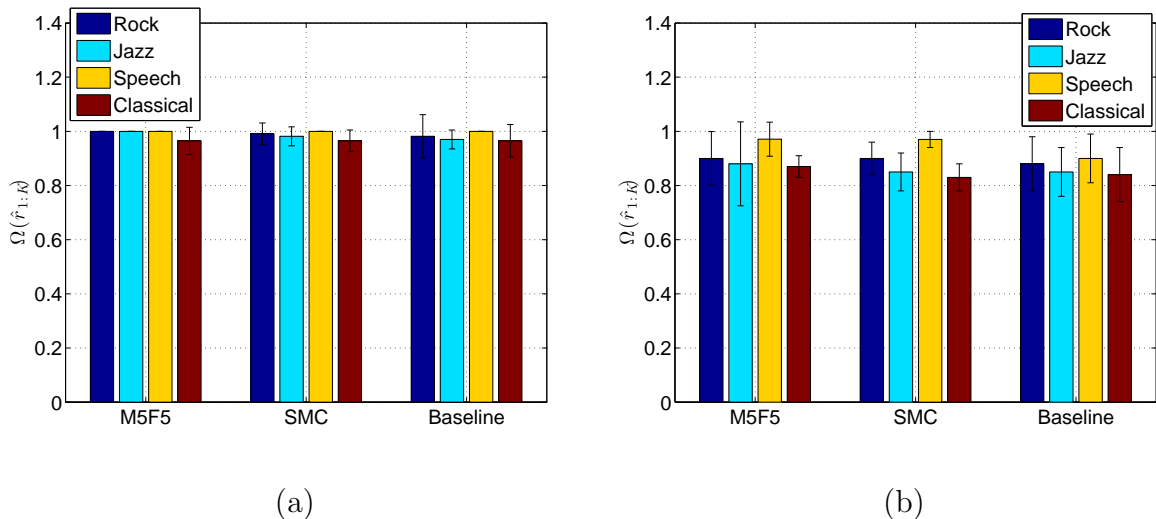


Figure 5.9. Alignment performances of sequential method with  $M_5F_5$ , SMC Sampler with  $\Phi_{M_5}(\mathbf{r})$  and baseline method in mean alignment scores and standard deviations for rock, jazz, classical music and speech a) for high SNR cases ( $\approx 10\text{dB}$ ) b) for low SNR cases ( $\approx -3\text{dB}$ ).

A final analysis is performed on the  $F_5$  feature for hashing algorithms. In [15], [31]

<sup>10</sup>Originally model  $M_5$  has 5 parameters  $\Theta = \{\alpha_\lambda, w_{0,0}, w_{1,0}, w_{0,1}, w_{1,1}\}$  however we simplify the model by assigning  $w = w_{1,1} = w_{0,0}$  and  $1 - w = w_{1,0} = w_{0,1}$

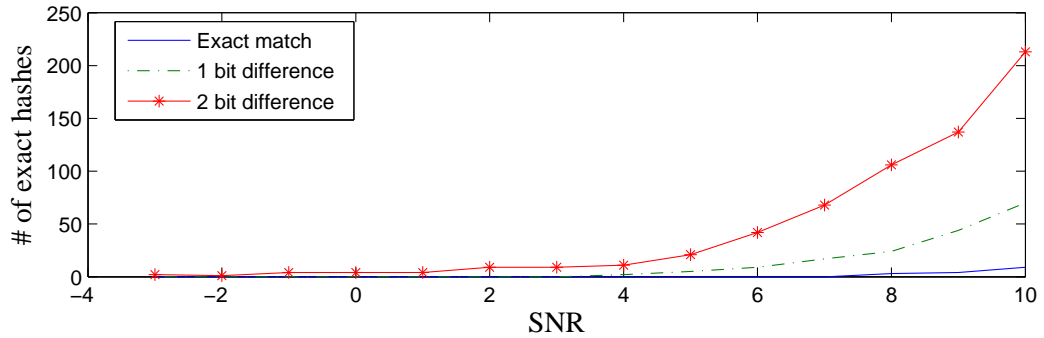


Figure 5.10. The number of exact, 1-bit difference and 2-bit difference hash matches between two noisy versions of the same song with varying SNR levels.

and [28],  $F_5$  feature is used to obtain a hash table where exact and near exact matches are searched. To test this issue, we used the following setup; two sequences are produced from one music file, for each sequence a different bar ambiance noise is added to the signal. The amplitudes of noises are adjusted to obtain different levels of SNR i.e., between -3dB and 10dB. The feature  $F_5$  is extracted from two sequences for each SNR level. Then, exact match, 1-bit difference and 2-bit differences are computed between columns of the sequences. The results are shown in Figure 5.10. As it can be observed, for small SNR values, there is no exact match, 1-bit and 2-bit differences occurred very rarely. Therefore we conclude that in this alignment setup, exact or near exact hashing strategies will not work with  $F_5$ .

### 5.5.3. Experiment 2: Real-life Data

In this experiment, we evaluate the algorithms for real use cases and compare with one of the baseline methods. The experiments are conducted on two large data sets. The data sets are described briefly below.

- *Data set 1*

The data set consists of 49 sequences from 6 different cameras. The number of clips recorded by each camera is listed in Table 5.7. In this case, more than one shot has been taken from each camera. The recordings are taken during a concert with a small audience (20-30 people) with varying qual-

ity recorders. There is no sound system and recordings are contaminated with conversational speech from the audience and other environmental sound sources. Total length of the data set is around 70 minutes. In the data set there are 8 isolated sequences, 6 pairs, 3 clusters of four sequences and 2 clusters of five sequences, 1 cluster of seven sequences.

- *Data set 2*

The data set consists of 113 sequences from 2 different cameras (51 recordings from recorder1 and 62 recordings from recorder 2). Again, more than one shot has been taken from each camera. The recordings are taken during a shooting of a commercial movie. The recordings are contaminated with conversational speech from the audience and other environmental sound sources. Total length of the data set is around 60 minutes. In the data set there are 32 isolated sequences, 39 pairs and 1 cluster of three sequences.

The ground-truth alignments for the data set 1 are obtained by careful listening. For data set 2, they are provided by the Singular Software Company.

Table 5.7. Number of clips taken by each camera for data set 1.

<b>Camera number</b>	Number of clips
Camera 1	6
Camera 2	11
Camera 3	7
Camera 4	11
Camera 5	5
Camera 6	9

The results are evaluated for model-feature pair  $M_5F_5$ , SMC Sampler with  $\Phi_{M_5}(\mathbf{r})$  and the baseline correlation method using both data sets and the alignment performances are compared according to their alignment score  $\Omega(\hat{r}_{1:K})$ . The alignment performances are listed in Table 5.8 for both data set 1 and data set 2.

In the feature extraction part for  $M_5F_5$  and SMC sampler, all audio sequences are time normalized to 16kHz, and 32 logarithmically spaced sub-bands are used as suggested in [15]. For the hyperparameter choice, we follow the values for low SNR cases given in Table 5.1 for  $M_5F_5$ . For the baseline correlation method, we tune the threshold for each data set separately to obtain best alignment performance. The threshold values are set as 0.62 and 0.9 for the data set 1 and data set 2, respectively.

In the SMC sampler implementation, the number of low resolution levels is chosen as nine and the minimum number of samples for low resolution bridge  $\Phi_L(\mathbf{r})$  is chosen as 40. For the alignment of  $k$ 'th sequence in the sequential algorithm, the lowest possible resolution is chosen according to the length of  $\Phi_L(\mathbf{r})$  or the length of the sequence  $N_k$ . Note that with each resolution level, the length of the sequence is halved and lowest possible length is 1. Hence the alignment of different sequences might start from different resolutions i.e. different number of samples. As an additional effort to improve the performance of the SMC sampler method, we apply a simulated annealing strategy on the hyperparameter  $w$  over the low resolution bridge functions  $\Phi_L(\mathbf{r})$  similar to annealing applied in Gibbs sampling approach in Section 5.5.1. The low resolution bridge functions act as coarse versions of actual score function  $\Phi_{M_5}(\mathbf{r})$ , which is equivalent to aligning noisier sequences for each low resolution level. As we choose smaller  $w$  values for noisy sequences, we start with a small  $w$  i.e.,  $w = 0.51$ , and gradually increase  $w$  with increasing resolution of bridge functions (Start from  $w = 0.51$  up to  $w = 0.62$ ).

There are four types of error in multiple alignment scenarios

Error type 1: A sequence  $\mathbf{x}_i$  that is not aligned to any other sequence at the ground truth (isolated), is aligned to a sequence  $\mathbf{x}_j$  in the alignment estimate.

Error type 2:  $\mathbf{x}_i$  that is aligned (overlap) to another sequence  $\mathbf{x}_j$  at the ground truth, is not aligned to any sequence in the alignment estimate.

Error type 3: A sequence  $\mathbf{x}_i$  that is aligned to the sequence  $\mathbf{x}_j$  at the ground truth, is aligned to another sequence  $\mathbf{x}_k$  in the alignment estimate.

Error type 4: A sequence  $\mathbf{x}_i$  that is aligned to the sequence  $\mathbf{x}_j$  at the ground

truth, is aligned to the same sequence with a wrong offset.

It is important to mention that error type 4 is not observed in the experiments with either of the data sets although it is observed in the experiments with the artificial data in Section 5.5.2.

Table 5.8. Alignment performances ( $\Omega(\hat{r}_{1:K})$ ) of model-feature pair  $M_5F_5$ , SMC Sampler with  $\Phi_{M_5}(\mathbf{r})$  and baseline correlation method for data set 1 and data set 2.

Alignment Method	$\Omega(\hat{r}_{1:K})$	
	Data Set 1	Data Set 2
$M_5F_5$	0.997	0.998
$SMC$	0.984	0.980
<i>Correlation</i>	0.959	0.82

Note that the clips that are recorded by the same device obviously do not overlap in the generic time line. In both data sets, all the cameras are used to record multiple clips. This brings an additional constraint on the estimation of alignments  $\hat{r}_{1:K}$  which reduces the search space for two reasons:

- (i) The sequences that are obtained from the same camera do not overlap hence alignment estimates  $\hat{r}_{1:K}$  that leads to overlapping between these sequences are not searched.
- (ii) If two sequences i.e.,  $\mathbf{x}_i$  and  $\mathbf{x}_j$  recorded by the cameras  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , respectively, are aligned (overlap) with each other, a sequence  $\mathbf{x}_k$  that is recorded by  $\mathcal{R}_1$  before  $\mathbf{x}_i$  do not overlap with any sequence  $\mathbf{x}_l$  that is recorded by  $\mathcal{R}_2$  after sequence  $\mathbf{x}_j$  and vice versa, then alignment estimates  $\hat{r}_{1:K}$  that leads to overlapping between sequences  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are not searched.

Reducing the search space decreases the computational time, however it might also lead to misalignments that affect the alignment performance  $\Omega(\hat{r}_{1:K})$  severely. This is due to the fact that if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  recorded by the cameras  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , respectively, are

misaligned with each other, the sequences recorded by  $\mathcal{R}_1$  before  $\mathbf{x}_i$  will not be searched for the sequences recorded by  $\mathcal{R}_2$  after  $\mathbf{x}_j$  and vice versa which might lead to severe decrease in the alignment performance. This is actually observed in the experiment where baseline correlation method is used to align data set 2, due to the two similar but not overlapping sequences (see Section 5.4 in false positive/negative analysis part).

Note that, even with such a misalignment, the alignment score for baseline method is not so low. This is due to the fact that if a data set that consists more of isolated sequences or clusters with few number of sequences, then possible misalignments would have a lower effect on the alignment performance on that data set. To illustrate this idea, we give three misalignment scenarios.

- Scenario 1: Assume that an isolated sequence  $\mathbf{x}_i$  at the ground truth, is aligned with a cluster  $\mathcal{C}_k$  of  $\mathcal{M}_k$  sequences such that it overlaps with each of the sequences in that cluster. Such a misalignment brings  $\mathcal{M}_k - 1$  errors on the alignment performance.
- Scenario 2: Assume that at the ground truth, one of the sequences  $\mathbf{x}_i$  in a cluster  $\mathcal{C}_k$  of  $\mathcal{M}_k$  sequences, overlaps with each other sequence in that cluster and it is not aligned to any cluster in the alignment estimate (decided as an isolated sequence). Such a misalignment brings  $\mathcal{M}_k - 1$  errors on the alignment performance.
- Scenario 3: Assume that at the ground truth, one of the sequences  $\mathbf{x}_i$  in a cluster  $\mathcal{C}_k$  of  $\mathcal{M}_k$  sequences, overlaps with each other sequence in that cluster and it is aligned to another cluster  $\mathcal{C}_l$  where it is overlapping with  $\mathcal{M}_l$  number of sequences in the alignment estimate. Such a misalignment brings  $\mathcal{M}_k - 1 + \mathcal{M}_l$  errors on the alignment performance.

It is important to mention that being recorded by the same camera does not mean that the recordings can not be in the same cluster but they just do not overlap. As an example, the cluster with 7 sequences in data set 1 includes two sequences from the 4th camera.

## 6. CONCLUSIONS AND FUTURE RESEARCH

In this thesis study, the multiple audio sequence alignment problem is addressed where the objective is to find the offset setting of  $K$  sequences on the generic time line under the circumstances that no prior information on offsets is available. This thesis proposes a novel, model based approach and a probabilistic interpretation for this problem. The model is quite generic and can be used for a variety of feature sets by choosing the observation model properly. Note that the observation and prior distributions chosen for this study is by no means exhaustive; any other conjugate prior pair of distributions is valid. To this end, six generative models are defined following the template model given in Chapter 2 and for each model we derived proper scoring functions by integrating over the latent variable  $\lambda$ .

In the alignment framework, emphasis is given on two issues: representation of the sequences and methods on relative offset search. Audio fingerprints which are binary by nature, are vastly used to represent audio signals due to their compact structure and robustness against degradations in audio. In this study, a wide range of feature types are investigated for the alignment task besides binary representations. Experiments under various degradations of audio show that binary feature  $F_5$  which is defined in [15] has most promising results. This feature is not only robust under different types of noise but also immune to volume variations between sequences since the signs of the differences in spectrogram are extracted. The next best feature sequence for the alignment task is determined as the positive spectral difference  $F_2$  which extracts the onsets of the audio signal.

In literature, hashing methods are widely used to find aligning sequences. These methods are computationally efficient but they rely on a simple assumption that some of the extracted fingerprints from two matching sequences would be exactly the same. This assumption might be true in audio matching setup where a short sequence is matched to large clean audio database. However due to the variability of sound quality and noise in each recording, an exact match in the fingerprints may not occur in the

alignment setup. Our experiments in Section 5.5.2 with popular fingerprint given in [15] (feature  $F_5$ ) support this claim such that for two low SNR sequences, exact match does not occur.

Other methods for offset search include simple similarity measures such as cross correlation and Hamming distance. The experimental results on real pairwise sets show that both measures have high alignment performances, however they lack of the ability to distinguish overlapping/non-overlapping sequences because these tools can not be used to quantify a non-overlapping result. On the other hand, the derived scoring functions  $\Phi(\mathbf{r})$  have resemblance with baseline similarity methods (see Section 3.1 and Section 3.2) however they are different in a way that they can penalize the non-overlapping sequences automatically without using some ad hoc thresholds.

Since the alignment is applied in the feature domain, the obtained results are up to a resolution of STFT hop-size and we apply no post-processing step afterwards. However in some applications, a finer alignment might be needed such as merging the raw signals to a single signal as in the concert recording scenario. One way to accomplish this, is decreasing the hop-size. But it would increase the size of the feature sequences and hence would increase the computational time. A feasible solution might be first working on a lower temporal resolution, then once the alignment is obtained, switching to a higher resolution alignment in the neighbourhood of the optimum alignment as proposed in Section 4.4.

It is important to mention that the model based approach is able to handle several types of signal distortions that may occur due to additive noise, recording device or channel effects. However one of the limitations in this work is the temporal deformations such as time warping. Multi sequence alignment with temporal deformations (both uniform and non-uniform) is a much more difficult problem to solve as the search space dramatically increases yet an important research topic to consider.

Two important criteria in the alignment framework are robustness and computational time. To test the robustness of the model, first model-feature pairs are exhaus-

tively tested with several experiments using artificially formed sequences, simulating the effect of volume variations, the amount of noise and the quality of recordings on the alignment performance. The results suggest that the model based approach has a high alignment performance even under very noisy conditions. Then to test the robustness of the model under real-life scenarios, we apply the  $M_5F_5$  model-feature pair and SMC sampler method to two real-life data sets compare the results to a correlation based baseline method. The results show that best alignment performance is obtained with  $M_5F_5$  however it has slightly higher performance than SMC sampler method. Both methods outperform the baseline method that supports the robustness of our approach.

It is important to mention that, among model-feature pairs,  $M_5F_5$  has significantly more promising results than others. For this reason, SMC sampler based multi resolution alignment method that is proposed in Section 4.4, is implemented using  $\Phi_{M_5}(\mathbf{r})$  and feature  $F_5$ .

Another issue that effects the robustness is the hyperparameter settings. As explained in Section 3.4, if the sequences have low SNR, the parameters should be chosen such that the precision of the observation model becomes low. It is observed that as the precision decreases, spurious peaks occur at the sides of scoring functions and the unreliable region becomes wider. Another interpretation is that as the sequences become noisier, the amount of data needed for a reliable alignment also increases. Hence the alignments for short sequences ( $< 3$  seconds) or sequences with small overlaps are not reliable in low SNR cases.

The main bottleneck with our approach is the high computational cost of the sequential search due to a complexity of  $O(N^{K-1})$ . The longest running time of our algorithm scales linearly with the total length of the feature sequences. In addition, inefficient computation of  $\Phi(r)$  functions lead to dramatic increase in computational time. The correlation based baseline method is much faster than both sequential alignment with  $M_5F_5$  and SMC sampler method due to this fact. Some suggestions and future research directions are discussed in Section 6.1.

## 6.1. Future Research Directions

For future research on multiple audio alignment, several directions are worthy of attempt.

Aforementioned, the main disadvantage of our approach is the computational complexity due to structure of score functions and the sequential search mechanism. Even multi resolution alignment integration to the sequential algorithm do not reduce the computational complexity enough. Hence, future research will be concentrated mostly on the improvement of computational performance of sequential algorithm.

One straightforward solution is parallelization which is possible since each alignment between a sequence and a cluster is computed independently. In addition, efficient computational methods could be employed for some of the  $\Phi(\mathbf{r})$  functions i.e., FFT-based techniques can be applied for  $\Phi_{M_3}(\mathbf{r})$ .

We observe in the experiments that the initial ordering of the sequences affects the computational time of the sequential algorithm dramatically. Therefore, a pre-knowledge on the matchings of sequences and a related initial ordering would certainly decrease computational time. For this purpose, a pre-classification of sequences according to some audio classes [28, 30] or noise type and level can be used as a preprocessing step. Then, sequences from the same class are more likely to match with each other. Another solution is to apply hashing methods similar to [1]. Depending on the level of noise, hashing methods would give an idea on the matchings up to some degree which can then be used as a preprocessing step to the sequential algorithm. In addition, we anticipate that ideally the methods described in this work and hashing strategies can be unified under a hybrid algorithm which might be both robust and fast for finding high resolution alignments.

Note that the multi dimensional  $\Phi(\mathbf{r})$  surface is very rough and standard optimization methods do not apply in this situation. Our preliminary experiments with batch Monte Carlo methods (Gibbs sampling), has not proven very effective even when

enhanced with different annealing techniques. Thus, we resort to the proposed heuristic sequential search algorithm. However more advanced sampling algorithms such as SMC samplers could be employed and a search mechanisms directly defined on the  $\Phi(\mathbf{r})$  could be developed. Therefore some effort should be given on research on sampling methods and how to apply these methods on  $\Phi(\mathbf{r})$  functions.

Experiments show that the models and the sequential alignment algorithm has a high performance both in artificial and real-life data sets. However, to further improve the performance of the algorithm, some effort should be given on hyperparameter choices. In this work, we defined two sets of parameters one for low SNR and one for high SNR cases. However in reality, the sequences in one data set may differ in noise level which may lead to misalignments i.e., error type 2 and type 3 may occur for not enough precision. As a result, an improvement on the performance can be obtained in the following way: An analysis of the current sequence or cluster against noise level will be conducted, then according to the levels corresponding hyperparameters will be used.

## APPENDIX A: DERIVATIONS OF SCORE FUNCTIONS

### $\Phi_{M_i}(\mathbf{r})$ FOR EACH MODEL

In this chapter, the derivations of score functions  $\Phi_{M_i}(r_{1:K})$  are given for each model ( $M_1, M_2, M_3, M_4, M_5$  and  $M_6$ ). Through out the derivations of each model, the probability of  $k$ 'th sequence to be aligned at time  $\tau$  is represented with  $\pi_{\tau,k}$ . Due to the fact that most of the time, there is no prior knowledge on the alignment of sequences, each  $r_k$  is assumed to be uniformly distributed. Note that each observation and prior distribution pairs are chosen conjugate prior so that the analytical derivation is possible.

#### A.1. Gamma Observation Model ( $M_1$ )

The model is,

$$\begin{aligned} \lambda_{\tau,f} &\sim \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \\ r_k &\sim \prod_{\tau=1}^{T-N_k+1} \pi_{\tau,k}^{[r_k=\tau]} \\ x_{k,n,f} | r_k, \lambda_{\tau,f} &\sim \prod_{\tau=1}^T \mathcal{G}(x_{k,n,f}; \alpha, \frac{\alpha}{\lambda_{\tau,f}})^{[n=\tau-r_k]} \end{aligned}$$

The exponential forms of the Gamma and inverse Gamma distributions are,

$$\begin{aligned} \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) &= \exp(-(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda) \\ \mathcal{G}(x_{k,n,f}; \alpha, \alpha/\lambda_{\tau,f}) &= \exp((\alpha - 1) \log x_{k,n,f} - \frac{\alpha}{\lambda_{\tau,f}} x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \frac{\alpha}{\lambda_{\tau,f}}) \end{aligned}$$

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{k,n,f} | r_k, \lambda_{1:T,f}) \prod_{f=1}^F \prod_{\tau=1}^T p(\lambda_{\tau,f}) \\
&= \prod_{f=1}^F \prod_{\tau=1}^T \exp\left(-(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right) \\
&\quad \prod_{k=1}^K \prod_{n=0}^{N_k-1} \exp\left([n = \tau - r_k] \left((\alpha - 1) \log x_{k,n,f} - \frac{\alpha}{\lambda_{\tau,f}} x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \frac{\alpha}{\lambda_{\tau,f}}\right)\right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left((\alpha - 1) \log x_{k,n,f} - \frac{\alpha}{\lambda_{\tau,f}} x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \frac{\alpha}{\lambda_{\tau,f}}\right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}}\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(-\frac{\alpha}{\lambda_{\tau,f}} x_{k,n,f} - \alpha \log \lambda_{\tau,f}\right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -(\alpha_\lambda + 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \log \lambda_{\tau,f}\right. \\
&\quad \quad \left. - (\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f}) \frac{1}{\lambda_{\tau,f}}\right)
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:K,0:N_{k-1},1:F}|r_{1:K}) &= \int d\lambda_{1:T,1:F} p(x_{1:K,0:N_{k-1},1:F}, \lambda_{1:T,1:F}|r_{1:K}) \\
&= \exp\left(-FT \log \Gamma(\alpha_\lambda) + FT \alpha_\lambda \log \beta_\lambda\right. \\
&\quad \left. + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] ((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha)\right) \\
&\int d\lambda_{1:T,1:F} \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -(\alpha_\lambda + 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \log \lambda_{\tau,f}\right. \\
&\quad \left. - (\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f}) \frac{1}{\lambda_{\tau,f}}\right) \\
&= \exp\left(-TF \log \Gamma(\alpha_\lambda) + TF \alpha_\lambda \log \beta_\lambda\right. \\
&\quad \left. + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] ((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha)\right) \\
&\int d\lambda_{1:T,1:F} \prod_{f=1}^F \prod_{\tau=1}^T IG(\lambda_{\tau,f}; \alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha \\
&\quad , \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f}) \\
&\exp(\log \Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \\
&\quad - (\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \log(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f})) \\
&= \exp(-TF \log \Gamma(\alpha_\lambda) + TF \alpha_\lambda \log \beta_\lambda \\
&\quad + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] ((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha) \\
&\quad + \sum_{f=1}^F \sum_{\tau=1}^T \log \Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \\
&\quad - \sum_{f=1}^F \sum_{\tau=1}^T (\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \log(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f}))
\end{aligned}$$

Then the score function  $\Phi_{M_1}(r_{1:K}) = \log p(x_{1:K,0:N_k-1}|r_{1:K})$  is obtained as,

$$\begin{aligned}
\Phi_{M_1}(r_{1:K}) &= -TF \log \Gamma(\alpha_\lambda) + TF \alpha_\lambda \log \beta_\lambda \\
&+ \sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] ((\alpha - 1) \log x_{k,n,f} - \log \Gamma(\alpha) + \alpha \log \alpha) \\
&+ F \sum_{\tau=1}^T \log \Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \\
&- \sum_{f=1}^F \sum_{\tau=1}^T (\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha) \\
&\quad \log(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \alpha x_{k,n,f})
\end{aligned} \tag{A.1}$$

## A.2. Gaussian Variance Observation Model ( $M_2$ )

The model is,

$$\begin{aligned}
\lambda_{\tau,f} &\sim \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \\
r_k &\sim \prod_{\tau=1}^T \pi_{k,\tau}^{[r_k=\tau]} \\
x_{k,n,f} | r_k, \lambda_{\tau,f} &\sim \prod_{\tau=1}^T \mathcal{N}(x_{k,n,f}; 0, \lambda_{\tau,f})^{[n=\tau-r_k]}
\end{aligned}$$

The exponential forms of the inverse gamma and Gaussian distributions are,

$$\begin{aligned}
\mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) &= \exp(-(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda) \\
\mathcal{N}(x_{k,n,f}; 0, \lambda_{\tau,f})^{[n=\tau-r_k]} &= \exp\left([n = \tau - r_k] \left(-\frac{x_{k,n,f}^2}{2\lambda_{\tau,f}} - \frac{1}{2} \log(2\pi\lambda_{\tau,f})\right)\right)
\end{aligned}$$

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{k,n,f} | r_k, \lambda_{1:T,f}) \prod_{f=1}^F \prod_{\tau=1}^T p(\lambda_{\tau,f}) \\
&= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{N}(x_{k,n,f}; 0, \lambda_{\tau,f})^{[n=\tau-r_k]} \prod_{\tau=1}^T \prod_{f=1}^F \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \\
&= \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{IG}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \prod_{k=1}^K \prod_{n=0}^{N_k-1} \mathcal{N}(x_{k,n,f}; 0, \lambda_{\tau,f})^{[n=\tau-r_k]} \\
&= \prod_{f=1}^F \prod_{\tau=1}^T \exp\left(-(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right) \\
&\quad \prod_{k=1}^K \prod_{n=0}^{N_k-1} \exp\left([n = \tau - r_k] \left(-\frac{x_{k,n,f}^2}{2\lambda_{\tau,f}} - \frac{1}{2} \log(2\pi \lambda_{\tau,f})\right)\right) \\
&= \exp\left[\sum_{f=1}^F \sum_{\tau=1}^T \left(-(\alpha_\lambda + 1) \log \lambda_{\tau,f} - \frac{\beta_\lambda}{\lambda_{\tau,f}} - \log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda\right.\right. \\
&\quad \left.\left. + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(-\frac{x_{k,n,f}^2}{2\lambda_{\tau,f}} - \frac{1}{2} \log 2\pi - \frac{1}{2} \log(\lambda_{\tau,f})\right)\right)\right]
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F} | r_{1:K}) &= \int d\lambda_{1:T,1:F} p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) \\
&= \exp\left(TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda)\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(-\frac{1}{2} \log 2\pi\right)\right) \\
&\quad \int d\lambda_{1:T,1:F} \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -(\alpha_\lambda + 1) \log \lambda_{\tau,f} + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2} \log(\lambda_{\tau,f})\right. \\
&\quad \left. + \sum_{f=1}^F \sum_{\tau=1}^T -(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}) \frac{1}{\lambda_{\tau,f}}\right)
\end{aligned}$$

$$\begin{aligned}
&= \exp\left(TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda)\right) \\
&\exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(-\frac{1}{2} \log 2\pi\right)\right) \\
&\int d\lambda_{1:T,1:F} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{IG}\left(\lambda_{\tau,f}; \alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right. \\
&\quad \left., \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}\right) \\
&\exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \log \Gamma\left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right)\right. \\
&\quad \left.- \left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right) \log\left(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}\right)\right) \\
&= \exp\left(TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda)\right) \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(-\frac{1}{2} \log 2\pi\right)\right) \\
&\exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \log \Gamma\left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right)\right) \\
&\exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -\left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right) \log\left(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}\right)\right) \\
&= \exp\left(TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda)\right) \exp\left(-F\left(\frac{1}{2} \log 2\pi\right) \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k]\right) \\
&\exp\left(F \sum_{\tau=1}^T \log \Gamma\left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right)\right) \\
&\exp\left(-\sum_{f=1}^F \sum_{\tau=1}^T \left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right) \log\left(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}\right)\right) \\
&= \exp\left(TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda)\right) \exp\left(-F\left(\frac{1}{2} \log 2\pi\right) \sum_{k=1}^K N_k\right) \\
&\exp\left(F \sum_{\tau=1}^T \log \Gamma\left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right)\right) \\
&\exp\left(-\sum_{f=1}^F \sum_{\tau=1}^T \left(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}\right) \log\left(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2}\right)\right)
\end{aligned}$$

Then the score function  $\Phi_{M_2}(r_{1:K}) = \log p(x_{1:K,0:N_k-1,1:F} | r_{1:K})$  is obtained as,

$$\begin{aligned}
\Phi_{M_2}(r_{1:K}) &= TF(-\log \Gamma(\alpha_\lambda) + \alpha_\lambda \log \beta_\lambda) - F\left(\frac{1}{2} \log 2\pi\right) \sum_{k=1}^K N_k \\
&+ F \sum_{\tau=1}^T \log \Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}) \\
&- \sum_{f=1}^F \sum_{\tau=1}^T (\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{1}{2}) \\
&\quad \log(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \frac{x_{k,n,f}^2}{2})
\end{aligned} \tag{A.2}$$

### A.3. Gaussian Mean Observation Model ( $M_3$ )

The model is,

$$\begin{aligned}
\lambda_{\tau,f} &\sim \mathcal{N}(\lambda_{\tau,f}; 0, \beta_\lambda) \\
r_k &\sim \prod_{\tau=1}^{T-N_k+1} \pi_{\tau,k}^{[r_k=\tau]} \\
x_{k,n,f} | r_k, \lambda_{\tau,f} &\sim \prod_{\tau=1}^T \mathcal{N}(x_{k,n,f}; \lambda_{\tau,f}, \frac{1}{\alpha})^{[n=\tau-r_k]}
\end{aligned}$$

The Gaussian distributions are written in exponential form as,

$$\begin{aligned}
\mathcal{N}(\lambda_{\tau,f}; 0, \beta_\lambda) &= \exp\left(-\frac{1}{2} \log(2\pi\beta_\lambda) - \frac{\lambda_{\tau,f}^2}{2\beta_\lambda}\right) \\
\mathcal{N}(x_{k,n,f}; \lambda_{\tau,f}, \frac{1}{\alpha})^{[n=\tau-r_k]} &= \exp\left([n = \tau - r_k] \left(-\frac{1}{2} \log(2\pi\frac{1}{\alpha}) - \frac{\alpha x_{k,n,f}^2}{2}\right.\right. \\
&\quad \left.\left.+ \lambda_{\tau,f} \alpha x_{k,n,f} - \frac{\alpha \lambda_{\tau,f}^2}{2}\right)\right)
\end{aligned}$$

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) &= \left( \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{k,n,f} | r_k, \lambda_{1:T,f}) \right) \left( \prod_{f=1}^F \prod_{\tau=1}^T p(\lambda_{\tau,f}) \right) \\
&= \left( \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{N}(x_{k,n,f}; \lambda_{\tau,f}, \frac{1}{\alpha})^{[n=\tau-r_k]} \right) \left( \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{N}(\lambda_{\tau,f}; 0, \beta_\lambda) \right) \\
&= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \left( \exp\left([n = \tau - r_k] \left( -\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \alpha \frac{x_{k,n,f}^2}{2} + \lambda_{\tau,f} \alpha x_{k,n,f} - \frac{\alpha \lambda_{\tau,f}^2}{2} \right) \right) \right) \\
&\quad \left( \prod_{f=1}^F \prod_{\tau=1}^T \exp\left(-\frac{1}{2} \log(2\pi\beta_\lambda) - \frac{\lambda_{\tau,f}^2}{2\beta_\lambda}\right) \right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) - \frac{\lambda_{\tau,f}^2}{2\beta_\lambda}\right) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \right. \right. \\
&\quad \left. \left. - \frac{\alpha x_{k,n,f}^2}{2} + \lambda_{\tau,f} \alpha x_{k,n,f} - \frac{\alpha \lambda_{\tau,f}^2}{2} \right) \right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right) \right) \right. \\
&\quad \left. \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{\lambda_{\tau,f}^2}{2\beta_\lambda}\right) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(\lambda_{\tau,f} \alpha x_{k,n,f} - \frac{\alpha \lambda_{\tau,f}^2}{2}\right) \right) \right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right) \right) \right. \\
&\quad \left. \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2\beta_\lambda} - \frac{\alpha}{2} \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \right) \lambda_{\tau,f}^2 \right. \right. \\
&\quad \left. \left. + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] (\lambda_{\tau,f} \alpha x_{k,n,f}) \right) \right)
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F}|r_{1:K}) &= \int d\lambda_{1:T,1:F} p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F}|r_{1:K}) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right)\right)\right) \\
&\int d\lambda_{1:T,1:F} \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2\beta_\lambda} - \frac{\alpha}{2} \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \lambda_{\tau,f}^2 \right.\right. \\
&\quad \left.\left. + \left(\sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \alpha x_{k,n,f}\right) \lambda_{\tau,f}\right)\right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right)\right)\right) \\
&\int d\lambda_{1:T,1:F} \mathcal{N}\left(\lambda_{\tau,f}; \frac{\sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] (\alpha x_{k,n,f})}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]}, \right. \\
&\quad \left. \frac{1}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]}\right) \\
&\exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \frac{1}{2} \frac{\left(\sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] (\alpha x_{k,n,f})\right)^2}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]}\right) \\
&= \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\frac{1}{2} \log(2\pi\beta_\lambda) + \frac{1}{2} \frac{\left(\sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] (\alpha x_{k,n,f})\right)^2}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]}\right.\right. \\
&\quad \left.\left. + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right)\right)\right)
\end{aligned}$$

Then the score function  $\Phi_{M_3}(r_{1:K}) = \log p(x_{1:K,0:N_k-1,1:F}|r_{1:K})$  is obtained as,

$$\begin{aligned}
\Phi_{M_3}(r_{1:K}) &= -\frac{TF}{2} \log(2\pi\beta_\lambda) + \sum_{f=1}^F \sum_{\tau=1}^T \left(\frac{1}{2} \frac{\left(\sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \alpha x_{k,n,f}\right)^2}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]}\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{T-N_k+1} [n = \tau - r_k] \left(-\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{k,n,f}^2}{2}\right) \tag{A.3}
\end{aligned}$$

For the case where there are only two sequences, the alignment of first sequence is fixed as  $r_1 = N_2 + 1$  and the length of the hidden sequence is set as  $T = N_1 + 2N_2 - 1$ .

As the optimum alignment search strategy, the log-likelihood scores are computed for all possible alignments of sequence 2 ( $r_2 = 1, 2, \dots, N_1 + N_2$ ). With this configuration, the only case where the sequences do not overlap is  $r_2 = 1$ . It is also important to note that the indicator function  $[.]$  acts as a kronecker delta function  $\delta[.]$ . Therefore the multiplication of indicator function with one of the sequence coefficients can be simplified using sampling property in the following manner;

$$\begin{aligned} \sum_{n=0}^{T-N_1+1} [n = \tau - r_1]x_{1,n,f} &= \sum_{n=0}^{T-N_1+1} [n = \tau - N_2 - 1]x_{1,n,f} \\ &= x_{1,\tau-N_2-1,f} = \begin{cases} x_{1,\tau-N_2-1,f} & \text{if } N_2 + 1 \leq \tau \leq N_1 + N_2; \\ 0 & \text{else.} \end{cases} \\ \sum_{n=0}^{T-N_1+1} [n = \tau - r_2]x_{2,n,f} &= x_{2,\tau-r_2,f} = \begin{cases} x_{2,\tau-r_2,f} & \text{if } r_2 \leq \tau \leq r_2 + N_2 - 1; \\ 0 & \text{else.} \end{cases} \end{aligned}$$

The score function  $\Phi_{M_3}(\mathbf{r})$  can then be simplified as,

$$\begin{aligned} \Phi_{M_3}(r_2) &= -\frac{TF}{2} \log(2\pi\beta_\lambda) \\ &+ \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{2} \frac{\left( \sum_{n=0}^{T-N_1+1} [n = \tau - r_1](\alpha x_{1,n,f}) + \sum_{n=0}^{T-N_2+1} [n = \tau - r_2](\alpha x_{2,n,f}) \right)^2}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^2 \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]} \right) \\ &+ \sum_{n=0}^{T-N_1+1} [n = \tau - r_1] \left( -\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{1,n,f}^2}{2} \right) \\ &+ \sum_{n=0}^{T-N_2+1} [n = \tau - r_2] \left( -\frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha x_{2,n,f}^2}{2} \right) \\ &= -\frac{TF}{2} \log(2\pi\beta_\lambda) - TF(N_1 + N_2) \frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \\ &+ \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{2} \frac{\left( \alpha x_{1,\tau-r_1,f} + \alpha x_{2,\tau-r_2,f} \right)^2}{\frac{1}{\beta_\lambda} + \alpha \sum_{k=1}^2 \sum_{n=0}^{T-N_k+1} [n = \tau - r_k]} - \frac{\alpha x_{1,\tau-r_1,f}^2}{2} - \frac{\alpha x_{2,\tau-r_2,f}^2}{2} \right) \end{aligned}$$

$$\begin{aligned}
& - \left( \frac{\alpha x_{1,\tau-r_1,f}^2}{2} - \frac{\alpha x_{2,\tau-r_2,f}^2}{2} \right) \\
= & - \frac{TF}{2} \log(2\pi\beta_\lambda) - TF(N_1 + N_2) \frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \\
& + \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{2} \frac{x_{1,\tau-N_2-1,f}^2}{\frac{1}{\alpha^2\beta_\lambda} + \frac{1}{\alpha}(1 + \sum_{n=0}^{T-N_2+1}[n = \tau - r_2])} - \frac{\alpha x_{1,\tau-N_2-1,f}^2}{2} \right) \\
& + \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{2} \frac{x_{2,\tau-r_2,f}^2}{\frac{1}{\alpha^2\beta_\lambda} + \frac{1}{\alpha}(1 + \sum_{n=0}^{T-N_1+1}[n = \tau - N_2 - 1])} - \frac{\alpha x_{2,\tau-r_2,f}^2}{2} \right) \\
& + \sum_{f=1}^F \sum_{\tau=1}^T \frac{x_{1,\tau-N_2-1,f} x_{2,\tau-r_2,f}}{\frac{1}{\alpha^2\beta_\lambda} + \frac{1}{\alpha}(\sum_{n=0}^{T-N_1+1}[n = \tau - N_2 - 1] + \sum_{n=0}^{T-N_2+1}[n = \tau - r_2])} \\
= & - \frac{TF}{2} \log(2\pi\beta_\lambda) - TF(N_1 + N_2) \frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \\
& + \frac{\alpha}{2} \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_2+1}[n = \tau - r_2])} - 1 \right) x_{1,\tau-N_2-1,f}^2 \\
& + \frac{\alpha}{2} \sum_{f=1}^F \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_1+1}[n = \tau - N_2 - 1])} - 1 \right) x_{2,\tau-r_2,f}^2 \\
& + \sum_{f=1}^F \sum_{\tau=1}^T \frac{x_{1,\tau-N_2-1,f} x_{2,\tau-r_2,f}}{\frac{1}{\alpha^2\beta_\lambda} + \frac{2}{\alpha}}
\end{aligned}$$

Resulting score function for a pairwise case becomes,

$$\begin{aligned}
\Phi_{M_3}(r_2) = & - \frac{TF}{2} \log(2\pi\beta_\lambda) - TF(N_1 + N_2) \frac{1}{2} \log\left(\frac{2\pi}{\alpha}\right) \\
& + \frac{\alpha}{2} \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_2+1}[n = \tau - r_2])} - 1 \right) \sum_{f=1}^F x_{1,\tau-N_2-1,f}^2 \\
& + \frac{\alpha}{2} \sum_{\tau=1}^T \left( \frac{1}{\frac{1}{\alpha\beta_\lambda} + (1 + \sum_{n=0}^{T-N_1+1}[n = \tau - N_2 - 1])} - 1 \right) \sum_{f=1}^F x_{2,\tau-r_2,f}^2 \\
& + \frac{1}{\frac{1}{\alpha^2\beta_\lambda} + \frac{2}{\alpha}} \sum_{f=1}^F \sum_{\tau=1}^T x_{1,\tau-N_2-1,f} x_{2,\tau-r_2,f} \tag{A.4}
\end{aligned}$$

#### A.4. Bernoulli Observation Model

The model is,

$$\begin{aligned}\lambda_{\tau,f} &\sim \mathcal{B}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \\ r_k &\sim \prod_{\tau=1}^T \pi_{k,\tau}^{[r_k=\tau]} \\ x_{k,n,f} | r_k, \lambda_{\tau,f} &\sim \prod_{\tau=1}^T \mathcal{BE}(x_{k,n,f}; \lambda_{\tau,f})^{[n=\tau-r_k]}\end{aligned}$$

The exponential forms of the Beta and Bernoulli distributions are,

$$\begin{aligned}\mathcal{B}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) &= \exp\left((\alpha_\lambda - 1) \log \lambda_{\tau,f} + (\beta_\lambda - 1) \log(1 - \lambda_{\tau,f}) + \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right) \\ \mathcal{BE}(x_{k,n,f}; \lambda_{\tau,f}) &= \exp\left(\log\left(\frac{\lambda_{\tau,f}}{1 - \lambda_{\tau,f}}\right)x_{k,n,f} + \log(1 - \lambda_{\tau,f})\right)\end{aligned}$$

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{k,n,f} | r_k, \lambda_{1:T,f}) \prod_{\tau=1}^T \prod_{f=1}^F p(\lambda_{\tau,f}) \\ &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{BE}(x_{k,n,f}; \lambda_{\tau,f})^{[n=\tau-r_k]} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{B}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \\ &= \prod_{\tau=1}^T \prod_{f=1}^F \mathcal{B}(\lambda_{\tau,f}; \alpha_\lambda, \beta_\lambda) \prod_{k=1}^K \prod_{n=0}^{N_k-1} \mathcal{BE}(x_{k,n,f}; \lambda_{\tau,f})^{[n=\tau-r_k]} \\ &= \prod_{\tau=1}^T \prod_{f=1}^F \exp\left((\alpha_\lambda - 1) \log \lambda_{\tau,f} + (\beta_\lambda - 1) \log(1 - \lambda_{\tau,f}) + \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right) \\ &\quad \prod_{k=1}^K \prod_{n=0}^{N_k-1} \exp\left([n = \tau - r_k] \left(\log\left(\frac{\lambda_{\tau,f}}{1 - \lambda_{\tau,f}}\right)x_{k,n} + \log(1 - \lambda_{\tau,f})\right)\right)\end{aligned}$$

$$\begin{aligned}
&= \exp\left(\sum_{\tau=1}^T \sum_{f=1}^F (\alpha_\lambda - 1) \log \lambda_{\tau,f} + (\beta_\lambda - 1) \log(1 - \lambda_{\tau,f}) + \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right. \\
&\quad \left.+ \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (\log(\frac{\lambda_{\tau,f}}{1 - \lambda_{\tau,f}}) x_{k,n,f} + \log(1 - \lambda_{\tau,f}))\right) \\
&= \exp\left(\sum_{\tau=1}^T \sum_{f=1}^F (\alpha_\lambda - 1) \log \lambda_{\tau,f} + (\beta_\lambda - 1) \log(1 - \lambda_{\tau,f}) + \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right. \\
&\quad \left.+ \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (\log(\lambda_{\tau,f}) x_{k,n,f} + (1 - x_{k,n,f}) \log(1 - \lambda_{\tau,f}))\right) \\
&= \exp\left(TF \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)} + \sum_{\tau=1}^T \sum_{f=1}^F (\alpha_\lambda - 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f}) \log \lambda_{\tau,f}\right. \\
&\quad \left.+ (\beta_\lambda - 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f})) \log(1 - \lambda_{\tau,f})\right)
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F} | r_{1:K}) &= \int d\lambda_{1:T,1:F} p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) \\
&= \exp\left(TF \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right) \\
&\int d\lambda_{1:T,1:F} \exp\left(\sum_{\tau=1}^T \sum_{f=1}^F (\alpha_\lambda - 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f}) \log \lambda_{\tau,f}\right. \\
&\quad \left.+ (\beta_\lambda - 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f})) \log(1 - \lambda_{\tau,f})\right)
\end{aligned}$$

Inside the integral, the equation actually is very similar to the beta distribution. Since the area under a distribution is 1, we try to make the equation beta distribution by adding extra elements in the equation as follows

$$\begin{aligned}
&= \exp\left(TF \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right) \\
&\int d\lambda_{1:T,1:F} \prod_{\tau=1}^T \prod_{f=1}^F \mathcal{B}\left(\lambda_{\tau,f}; \alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f} \right. \\
&\quad \left. , \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f})\right) \\
&\exp\left(\sum_{\tau=1}^T \sum_{f=1}^F \log(\Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f}))\right) \\
&+ \log(\Gamma(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f}))) \\
&- \log(\Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f} + \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f}))) \\
&= \exp\left(TF \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)}\right) \\
&\exp\left(\sum_{\tau=1}^T \sum_{f=1}^F \log(\Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f}))\right) \\
&+ \log(\Gamma(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f}))) \\
&+ \log(\Gamma(\alpha_\lambda + \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k]))
\end{aligned}$$

Then the score function  $\Phi_{M_4}(r_{1:K}) = \log p(x_{1:K,0:N_k-1,1:F} | r_{1:K})$  is obtained as,

$$\begin{aligned}
\Phi_{M_4}(r_{1:K}) &= TF \log \frac{\Gamma(\alpha_\lambda + \beta_\lambda)}{\Gamma(\alpha_\lambda)\Gamma(\beta_\lambda)} \\
&- \sum_{\tau=1}^T \sum_{f=1}^F \log \Gamma(\alpha_\lambda + \beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k]) \\
&+ \sum_{\tau=1}^T \sum_{f=1}^F \log \Gamma(\alpha_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{k,n,f}) \\
&+ \sum_{\tau=1}^T \sum_{f=1}^F \log \Gamma(\beta_\lambda + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] (1 - x_{k,n,f})) \tag{A.5}
\end{aligned}$$

### A.5. Conditional Bernoulli Observation Model ( $M_5$ )

The model is,

$$\begin{aligned}\lambda_{\tau,f} &\sim \mathcal{BE}(\lambda_{\tau,f}; \alpha_\lambda) \\ r_k &\sim \prod_{\tau=1}^T \pi_{k,\tau}^{[r_k=\tau]} \\ x_{k,n,f}|r_k, \lambda_{\tau,f} &\sim \prod_{\tau=1}^T \mathcal{P}(x_{k,n,f}|r_k, \lambda_{1:T,f})^{[n=\tau-r_k]}\end{aligned}$$

The exponential forms of the Bernoulli and observation distribution are;

$$\begin{aligned}\mathcal{BE}(\lambda_{\tau,f}; \alpha_\lambda) &= \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\ \mathcal{P}(x_{k,n,f}|r_k, \lambda_{1:T,f}) &= \exp\left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{k,n,f} = i][\lambda_{\tau,f} = j] \log(w_{i,j})\right)\end{aligned}$$

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F}|r_{1:K}) &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{k,n,f}|r_k, \lambda_{1:T,f}) \prod_{\tau=1}^T \prod_{f=1}^F p(\lambda_{\tau,f}) \\ &= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{P}(x_{k,n,f}|r_k, \lambda_{1:T,f})^{[n=\tau-r_k]} \prod_{\tau=1}^T \prod_{f=1}^F \mathcal{BE}(\lambda_{\tau,f}; \alpha_\lambda) \\ &= \prod_{\tau=1}^T \prod_{f=1}^F \mathcal{BE}(\lambda_{\tau,f}; \alpha_\lambda) \prod_{k=1}^K \prod_{n=0}^{N_k-1} \mathcal{P}(x_{k,n,f}|r_k, \lambda_{1:T,f})^{[n=\tau-r_k]} \\ &= \prod_{\tau=1}^T \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\ &\quad \prod_{k=1}^K \prod_{n=0}^{N_k-1} \exp\left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{k,n,f} = i][\lambda_{\tau,f} = j] \log(w_{i,j})\right)^{[n=\tau-r_k]}\end{aligned}$$

$$\begin{aligned}
&= \exp\left(\sum_{\tau=1}^T \sum_{f=1}^F \log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{k,n,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right) \\
&= \prod_{\tau=1}^T \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \sum_{k=1}^K \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{k,\tau-r_k,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right)
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:K,0:N_k-1,1:F} | r_{1:K}) &= \sum_{\lambda_{1:T,1:F}} p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) \\
&= \sum_{\lambda_{1:T,1:F}} \prod_{\tau=1}^T \prod_{f=1}^F \exp\left(\log(\alpha_\lambda) \lambda_{\tau,f} + \log(1-\alpha_\lambda)(1-\lambda_{\tau,f})\right) \\
&\quad + \sum_{k=1}^K \sum_{i=0}^1 \sum_{j=0}^1 [x_{k,\tau-r_k,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j}) \\
&= \prod_{\tau=1}^T \prod_{f=1}^F \sum_{\lambda_{\tau,f}=0}^1 \exp\left(\log(\alpha_\lambda) \lambda_{\tau,f} + \log(1-\alpha_\lambda)(1-\lambda_{\tau,f})\right) \\
&\quad + \sum_{k=1}^K \sum_{i=0}^1 \sum_{j=0}^1 [x_{k,\tau-r_k,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j}) \\
&= \prod_{\tau=1}^T \prod_{f=1}^F \left( \exp\left(\log(1-\alpha_\lambda) + \sum_{k=1}^K \sum_{i=0}^1 [x_{k,\tau-r_k,f} = i] \log(w_{i,0})\right) \right. \\
&\quad \left. + \exp\left(\log(\alpha_\lambda) + \sum_{k=1}^K \sum_{i=0}^1 [x_{k,\tau-r_k,f} = i] \log(w_{i,1})\right) \right) \\
&= \prod_{\tau=1}^T \prod_{f=1}^F \left( \exp\left(\log(1-\alpha_\lambda) + \sum_{k=1}^K ([x_{k,\tau-r_k,f} = 0] \log(w_{0,0}) + [x_{k,\tau-r_k,f} = 1] \log(w_{1,0}))\right) \right. \\
&\quad \left. + \exp\left(\log(\alpha_\lambda) + \sum_{k=1}^K ([x_{k,\tau-r_k,f} = 0] \log(w_{0,1}) + [x_{k,\tau-r_k,f} = 1] \log(w_{1,1}))\right) \right)
\end{aligned}$$

If we choose the  $w_{1,0} = w_1$  and  $w_{0,1} = w_2$ , then  $w_{0,0} = 1 - w_1$  and  $w_{1,1} = 1 - w_2$  the equation becomes,

$$\begin{aligned} & \prod_{\tau=1}^T \prod_{f=1}^F \left( \exp \left( \log(1 - \alpha_\lambda) + \sum_{k=1}^K ([x_{k,\tau-r_k,f} = 0] \log(1 - w_1)) + [x_{k,\tau-r_k,f} = 1] \log(w_1) \right) \right) \\ & \quad + \exp \left( \log(\alpha_\lambda) + \sum_{k=1}^K ([x_{k,\tau-r_k,f} = 0] \log(w_2)) + [x_{k,\tau-r_k,f} = 1] \log(1 - w_2) \right) \Big) \\ & = \prod_{\tau=1}^T \prod_{f=1}^F \left( (1 - \alpha_\lambda)(1 - w_1)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} w_1^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right. \\ & \quad \left. + \alpha_\lambda w_2^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} (1 - w_2)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right) \end{aligned}$$

By further choosing  $w_{1,0} = w_{0,1} = w$ ,

$$\begin{aligned} p(x_{1:K,0:N_k-1,1:F} | r_{1:K}) & = \prod_{\tau=1}^T \prod_{f=1}^F \left( (1 - \alpha_\lambda)(1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right. \\ & \quad \left. + \alpha_\lambda w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} (1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right) \end{aligned}$$

Then the score function  $\Phi_{M_5}(r_{1:K}) = \log p(x_{1:K,0:N_k-1,1:F} | r_{1:K})$  is obtained as,

$$\begin{aligned} \Phi_{M_5}(r_{1:K}) & = \log \left( \prod_{\tau=1}^T \prod_{f=1}^F \left( (1 - \alpha_\lambda)(1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right. \right. \\ & \quad \left. \left. + \alpha_\lambda w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} (1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right) \right) \\ & = \sum_{\tau=1}^T \sum_{f=1}^F \log \left( \left( (1 - \alpha_\lambda)(1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right. \right. \\ & \quad \left. \left. + \alpha_\lambda w^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=0]} (1 - w)^{\sum_{k=1}^K [x_{k,\tau-r_k,f}=1]} \right) \right) \quad (\text{A.6}) \end{aligned}$$

For the case where there are only two sources, we set  $r_1 = N_2 + 1$  and the length of the hidden sequence as  $T = N_1 + 2N_2 - 1$ . As the optimum alignment search strategy, the log-likelihood scores are computed for all possible alignments of sequence

2 ( $r_2 = 1, 2, \dots, N_1 + N_2$ ). With this configuration, the only case where the sequences do not overlap is  $r_2 = 1$ . It is also important to note that the indicator function  $[\cdot]$  acts as a kronecker delta function  $\delta[\cdot]$ . Therefore the multiplication of indicator function with one of the sequence coefficients can be simplified using sampling property in the following manner;

$$\begin{aligned} \sum_{n=0}^{T-N_1+1} [n = \tau - r_1]x_{1,n,f} &= \sum_{n=0}^{T-N_1+1} [n = \tau - N_2 - 1]x_{1,n,f} \\ &= x_{1,\tau-N_2-1,f} = \begin{cases} x_{1,\tau-N_2-1,f} & \text{if } N_2 + 1 \leq \tau \leq N_1 + N_2; \\ 0 & \text{else.} \end{cases} \\ \sum_{n=0}^{T-N_1+1} [n = \tau - r_2]x_{2,n,f} &= x_{2,\tau-r_2,f} = \begin{cases} x_{2,\tau-r_2,f} & \text{if } r_2 \leq \tau \leq r_2 + N_2 - 1; \\ 0 & \text{else.} \end{cases} \end{aligned}$$

By choosing  $\alpha_\lambda = 0.5$ ,  $w_{0,0} = w_{1,1} = w$  and  $w_{1,0} = w_{0,1} = 1 - w$ , the score function  $\Phi_{M_5}(\mathbf{r})$  can then be simplified as,

$$\begin{aligned} \Phi_{M_5}(r_2) &= \sum_{\tau=1}^T \sum_{f=1}^F \log \left( 0.5(1-w)^{([x_{1,\tau-N_2-1,f}=0]+[x_{2,\tau-r_2,f}=0])} w^{([x_{1,\tau-N_2-1,f}=1]+[x_{2,\tau-r_2,f}=1])} \right. \\ &\quad \left. + 0.5w^{([x_{1,\tau-N_2-1,f}=0]+[x_{2,\tau-r_2,f}=0])} (1-w)^{([x_{1,\tau-N_2-1,f}=1]+[x_{2,\tau-r_2,f}=1])} \right) \end{aligned} \quad (\text{A.7})$$

## A.6. Conditional Bernoulli Observation Model - Alternative Approach for Pairwise Alignment

In this approach, it is assumed that there are two observed sequences. The model is some parts of the hidden sequence i.e.,  $\lambda_{N_2+1:N_1+N_2,1:F}$ , and one sequence  $x_{0:N_2-1,1:F}$

is observed. Then the joint conditional distribution is,

$$\begin{aligned}
p(x_{0:N_2-1,1:F}, \lambda_{1:T,1:F} | r) &= \exp\left(\sum_{\tau=1}^T \sum_{f=1}^F \log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \sum_{n=0}^{N_2-1} [n = \tau - r] \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{n,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right) \\
&= \prod_{\tau=1}^T \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda) + \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right)\right)
\end{aligned}$$

Then integrating out the latent variables that are not observed namely,  $\lambda_{1:N_2,1:F}$  and  $\lambda_{N_1+N_2+1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{0:N_2-1,1:F} | r_{1:K}) &= \sum_{\lambda_{1:N_2,1:F}} \sum_{\lambda_{N_1+N_2+1:T,1:F}} p(x_{1:K,0:N_k-1,1:F}, \lambda_{1:T,1:F} | r_{1:K}) \\
&= \sum_{\lambda_{1:N_2,1:F}} \sum_{\lambda_{N_1+N_2+1:T,1:F}} \prod_{\tau=1}^T \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right) \\
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})\right) \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \sum_{\lambda_{\tau,f}=0}^1 \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right) \lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad + \sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f} = i] [\lambda_{\tau,f} = j] \log(w_{i,j})
\end{aligned}$$

$$\begin{aligned}
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda) + \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f}=i][\lambda_{\tau,f}=j] \log(w_{i,j})\right)\right) \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left(\exp\left(\log(1-\alpha_\lambda) + \sum_{i=0}^1 [x_{\tau-r,f}=i] \log(w_{i,0})\right)\right. \\
&\quad \left. + \exp\left(\log(\alpha_\lambda) + \sum_{i=0}^1 [x_{\tau-r,f}=i] \log(w_{i,1})\right)\right) \\
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda) + \left(\sum_{i=0}^1 \sum_{j=0}^1 [x_{\tau-r,f}=i][\lambda_{\tau,f}=j] \log(w_{i,j})\right)\right) \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left(\exp\left(\log(1-\alpha_\lambda) + [x_{\tau-r,f}=0] \log(w_{0,0}) + [x_{\tau-r,f}=1] \log(w_{1,0})\right)\right. \\
&\quad \left. + \exp\left(\log(\alpha_\lambda) + [x_{\tau-r,f}=0] \log(w_{0,1}) + [x_{\tau-r,f}=1] \log(w_{1,1})\right)\right) \\
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right. \\
&\quad + [x_{\tau-r,f}=0][\lambda_{\tau,f}=0] \log(w_{0,0}) \\
&\quad + [x_{\tau-r,f}=1][\lambda_{\tau,f}=0] \log(w_{1,0}) \\
&\quad + [x_{\tau-r,f}=0][\lambda_{\tau,f}=1] \log(w_{0,1}) \\
&\quad \left. + [x_{\tau-r,f}=1][\lambda_{\tau,f}=1] \log(w_{1,1})\right) \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left(\left(1-\alpha_\lambda\right)w_{0,0}^{[x_{\tau-r,f}=0]}w_{1,0}^{[x_{\tau-r,f}=1]} + \alpha_\lambda w_{0,1}^{[x_{\tau-r,f}=0]}w_{1,1}^{[x_{\tau-r,f}=1]}\right) \\
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad w_{0,0}^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=0]}w_{1,0}^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=0]}w_{0,1}^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=1]}w_{1,1}^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=1]} \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left(\left(1-\alpha_\lambda\right)w_{0,0}^{[x_{\tau-r,f}=0]}w_{1,0}^{[x_{\tau-r,f}=1]} + \alpha_\lambda w_{0,1}^{[x_{\tau-r,f}=0]}w_{1,1}^{[x_{\tau-r,f}=1]}\right)
\end{aligned}$$

If we choose the  $w_{1,0} = w_1$  and  $w_{0,1} = w_2$ , then  $w_{0,0} = 1 - w_1$  and  $w_{1,1} = 1 - w_2$ , the equation becomes,

$$\begin{aligned}
&= \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \\
&\quad (1-w_1)^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=0]} w_1^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=0]} \\
&\quad w_2^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=1]} (1-w_2)^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=1]} \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left( (1-\alpha_\lambda)(1-w_1)^{[x_{\tau-r,f}=0]} w_1^{[x_{\tau-r,f}=1]} + \alpha_\lambda w_2^{[x_{\tau-r,f}=0]} (1-w_2)^{[x_{\tau-r,f}=1]} \right)
\end{aligned}$$

Then the log-likelihood is equal to,

$$\begin{aligned}
&= \log\left( \prod_{\tau=N_2+1}^{N_1+N_2} \prod_{f=1}^F \exp\left(\log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda)\right) \right. \\
&\quad (1-w_1)^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=0]} w_1^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=0]} \\
&\quad \left. w_2^{[x_{\tau-r,f}=0][\lambda_{\tau,f}=1]} (1-w_2)^{[x_{\tau-r,f}=1][\lambda_{\tau,f}=1]} \right) \\
&\prod_{\tau=1}^{N_2} \prod_{\tau=N_1+N_2+1}^T \prod_{f=1}^F \left( (1-\alpha_\lambda)(1-w_1)^{[x_{\tau-r,f}=0]} w_1^{[x_{\tau-r,f}=1]} + \alpha_\lambda w_2^{[x_{\tau-r,f}=0]} (1-w_2)^{[x_{\tau-r,f}=1]} \right) \\
&= \sum_{\tau=N_2+1}^{N_1+N_2} \sum_{f=1}^F \left( \log\left(\frac{\alpha_\lambda}{1-\alpha_\lambda}\right)\lambda_{\tau,f} + \log(1-\alpha_\lambda) \right. \\
&\quad + [x_{\tau-r,f}=0][\lambda_{\tau,f}=0] \log(1-w_1) + [x_{\tau-r,f}=1][\lambda_{\tau,f}=0] \log(w_1) \\
&\quad \left. + [x_{\tau-r,f}=0][\lambda_{\tau,f}=1] \log(w_2) + [x_{\tau-r,f}=1][\lambda_{\tau,f}=1] \log(1-w_2) \right) \\
&+ \sum_{\tau=1}^{N_2} \sum_{\tau=N_1+N_2+1}^T \sum_{f=1}^F \log\left( (1-\alpha_\lambda)(1-w_1)^{[x_{\tau-r,f}=0]} w_1^{[x_{\tau-r,f}=1]} \right. \\
&\quad \left. + \alpha_\lambda w_2^{[x_{\tau-r,f}=0]} (1-w_2)^{[x_{\tau-r,f}=1]} \right)
\end{aligned}$$

Choosing  $w_1 = w_2 = w$ ,  $\alpha_\lambda = 0.5$  and defining  $\mathcal{S}(r, \tau)$  as,

$$\mathcal{S}(r_2, \tau) = \begin{cases} 1 & \text{if } x \text{ exists at time } \tau; \\ 0 & \text{else.} \end{cases}$$

The log-likelihood becomes,

$$\begin{aligned} &= \sum_{\tau=N_2+1}^{N_1+N_2} \sum_{f=1}^F \left( [x_{\tau-r,f} = 0][\lambda_{\tau,f} = 0] \log(1-w) + [x_{\tau-r,f} = 1][\lambda_{\tau,f} = 0] \log(w) \right. \\ &\quad \left. + [x_{\tau-r,f} = 0][\lambda_{\tau,f} = 1] \log(w) + [x_{\tau-r,f} = 1][\lambda_{\tau,f} = 1] \log(1-w) \right) \\ &+ \sum_{\tau=1}^{N_2} \sum_{\tau=N_1+N_2+1}^T \sum_{f=1}^F \log \left( 0.5(1-w)^{[x_{\tau-r,f}=0]} w^{[x_{\tau-r,f}=1]} + 0.5w^{[x_{\tau-r,f}=0]} (1-w)^{[x_{\tau-r,f}=1]} \right) \\ &= \sum_{\tau=N_2+1}^{N_1+N_2} \sum_{f=1}^F \left( [x_{\tau-r,f} = 0][\lambda_{\tau,f} = 0] \log(1-w) + [x_{\tau-r,f} = 1][\lambda_{\tau,f} = 0] \log(w) \right) \\ &+ \sum_{\tau=1}^{N_2} \sum_{\tau=N_1+N_2+1}^T \sum_{f=1}^F \log(0.5) \mathcal{S}(r, \tau) \end{aligned}$$

The resulting score function becomes,

$$\begin{aligned} \Phi_{M_{5.1}}(r) &= \sum_{\tau=N_2+1}^{N_1+N_2} \sum_{f=1}^F \left( ([x_{\tau-r,f} = 0][\lambda_{\tau,f} = 0]) \log(1-w) \right. \\ &\quad + ([x_{\tau-r,f} = 1][\lambda_{\tau,f} = 1]) \log(1-w) \\ &\quad + ([x_{\tau-r,f} = 0][\lambda_{\tau,f} = 1]) \log(w) \\ &\quad \left. + ([x_{\tau-r,f} = 1][\lambda_{\tau,f} = 0]) \log(w) \right) \\ &+ F \log(0.5) \sum_{\tau=1}^{N_2} \sum_{\tau=N_1+N_2+1}^T \mathcal{S}(r, \tau) \end{aligned} \tag{A.8}$$

### A.7. Multinomial Observation Model ( $M_6$ )

The model is,

$$\begin{aligned}\lambda_{1:Q,\tau,f} &\sim \text{Dir}(\lambda_{1:Q,\tau,f}; \alpha_{1:Q}) \\ r_k &\sim \prod_{\tau=1}^{T-N_k+1} \pi_{k,\tau}^{[r_k=\tau]} \\ x_{1:Q,k,n,f} | r_k, \lambda_{1:Q,\tau,f} &\sim \prod_{\tau=1}^T \mathcal{M}(x_{1:Q,k,n,f}; \lambda_{1:Q,\tau,f})^{[n=\tau-r_k]}\end{aligned}$$

where  $Q$  is the number of quantization levels.

The exponential forms of the dirichlet and multinomial distributions are

$$\begin{aligned}\text{Dir}(\lambda_{1:Q,\tau,f}; \alpha_{1:Q}) &= \frac{\Gamma(\sum_{q=1}^Q \alpha_q)}{\prod_{q=1}^Q \Gamma(\alpha_q)} \prod_{q=1}^Q (\lambda_{q,\tau,f})^{\alpha_q-1} \\ &= \exp\left(\sum_{q=1}^Q (\alpha_q - 1) \log \lambda_{q,\tau,f} + \log \Gamma(\sum_{q=1}^Q \alpha_q) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right) \\ \mathcal{M}(x_{1:Q,k,n,f}; 1, \lambda_{1:Q,\tau,f}) &= \lambda_{1,\tau,f}^{x_{1,k,n,f}} \lambda_{2,\tau,f}^{x_{2,k,n,f}} \dots \lambda_{Q,\tau,f}^{x_{Q,k,n,f}} \\ &= \exp\left(x_{1,k,n,f} \log \lambda_{1,\tau,f} + x_{2,k,n,f} \log \lambda_{2,\tau,f} + \dots + x_{Q,k,n,f} \log \lambda_{Q,\tau,f}\right) \\ &= \exp\left(\sum_{q=1}^Q \log \lambda_{q,\tau,f}^{x_{q,f,n,k}}\right) \\ &= \exp\left(\sum_{q=1}^Q x_{q,f,n,k} \log \lambda_{q,\tau,f}\right)\end{aligned}$$

Note that the multinomial distribution has the number of trial parameter as 1. Then the  $x_{1:Q,f,n,k}$  is a vector for which only one element of the vector is active and the rest of the elements are equal to zero. As an example, if there are  $Q=3$  levels and the second level is selected, the vector is,  $x_{1:Q,f,n,k} = \{0, 1, 0\}$ .

From the model, the conditional joint distribution can be written as,

$$\begin{aligned}
& p(x_{1:Q,1:K,0:N_k-1,1:F} | \lambda_{1:Q,1:T,1:F}, r_{1:K}) \\
&= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F p(x_{1:Q,k,n,f} | r_k, \lambda_{1:Q,1:T,f}) \prod_{\tau=1}^T \prod_{f=1}^F p(\lambda_{1:Q,\tau,f}) \\
&= \prod_{k=1}^K \prod_{n=0}^{N_k-1} \prod_{f=1}^F \prod_{\tau=1}^T \mathcal{M}(x_{1:Q,k,n,f}; 1, \lambda_{1:Q,\tau,f})^{[n=\tau-r_k]} \prod_{\tau=1}^T \prod_{f=1}^F \text{Dir}(\lambda_{1:Q,\tau,f}; \alpha_{1:Q}) \\
&= \prod_{f=1}^F \prod_{\tau=1}^T \text{Dir}(\lambda_{1:Q,\tau,f}; \alpha_{1:Q}) \prod_{k=1}^K \prod_{n=0}^{N_k-1} \mathcal{M}(x_{1:Q,k,n,f}; 1, \lambda_{1:Q,\tau,f})^{[n=\tau-r_k]} \\
&= \prod_{f=1}^F \prod_{\tau=1}^T \exp\left(\sum_{q=1}^Q (\alpha_q - 1) \log \lambda_{q,\tau,f} + \log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right) \\
&\quad \prod_{k=1}^K \prod_{n=0}^{N_k-1} \exp\left([n = \tau - r_k] \left(\sum_{q=1}^Q x_{q,f,n,k} \log \lambda_{q,\tau,f}\right)\right) \\
&= \exp\left[\sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q (\alpha_q - 1) \log \lambda_{q,\tau,f} + \log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right. \\
&\quad \left. + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \left(\sum_{q=1}^Q x_{q,f,n,k} \log \lambda_{q,\tau,f}\right)\right]
\end{aligned}$$

Then integrating out the latent variables  $\lambda_{1:Q,1:T,1:F}$  from the conditional joint distribution,

$$\begin{aligned}
p(x_{1:Q,1:K,0:N_k-1,1:F} | r_{1:K}) &= \exp\left(TF \left(\log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right)\right) \\
&\int d_{\lambda_{1:Q,1:T,1:F}} \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q (\alpha_q - 1 + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}) \log \lambda_{q,\tau,f}\right)
\end{aligned}$$

$$\begin{aligned}
&= \exp\left(TF\left(\log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right)\right) \\
&\quad \int d\lambda_{1:Q,1:T,1:F} \prod_{f=1}^F \prod_{\tau=1}^T \text{Dir}\left(\lambda_{1:Q,\tau,f}; \alpha_{1:Q} + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{1:Q,f,n,k}\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \left(-\log \Gamma\left(\sum_{q=1}^Q [\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}]\right)\right)\right) \\
&\quad + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q \log \Gamma\left(\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}\right) \\
&= \exp\left(TF\left(\log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right)\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T -\log \Gamma\left(\sum_{q=1}^Q [\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}]\right)\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q \log \Gamma\left(\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}\right)\right) \\
&= \exp\left(TF\left(\log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right)\right) \\
&\quad \exp\left(-\sum_{f=1}^F \sum_{\tau=1}^T \log \Gamma\left(\sum_{q=1}^Q \alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \sum_{q=1}^Q x_{q,f,n,k}\right)\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q \log \Gamma\left(\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}\right)\right) \\
&= \exp\left(TF\left(\log \Gamma\left(\sum_{q=1}^Q \alpha_q\right) - \sum_{q=1}^Q \log \Gamma(\alpha_q)\right)\right) \\
&\quad \exp\left(-F \sum_{\tau=1}^T \log \Gamma\left(\sum_{q=1}^Q \alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k]\right)\right) \\
&\quad \exp\left(\sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q \log \Gamma\left(\alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k}\right)\right)
\end{aligned}$$

Then the score function  $\Phi_{M_6}(r_{1:K}) = \log p(x_{1:Q,1:K,0:N_k-1,1:F}|r_{1:K})$  is obtained as,

$$\begin{aligned}
\Phi_{M_6}(r_{1:K}) &= TF \left( \log \Gamma \left( \sum_{q=1}^Q \alpha_q \right) - \sum_{q=1}^Q \log \Gamma(\alpha_q) \right) \\
&\quad - F \sum_{\tau=1}^T \log \Gamma \left( \sum_{q=1}^Q \alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] \right) \\
&\quad + \sum_{f=1}^F \sum_{\tau=1}^T \sum_{q=1}^Q \log \Gamma \left( \alpha_q + \sum_{k=1}^K \sum_{n=0}^{N_k-1} [n = \tau - r_k] x_{q,f,n,k} \right) \quad (\text{A.9})
\end{aligned}$$

## REFERENCES

1. Kennedy, L. and M. Naaman, “Less Talk, More Rock: Automated Organization of Community-Contributed Collections of Concert Videos”, *Proceedings of the 18th International Conference on World Wide Web*, pp. 311–320, 2009.
2. Cotton, C. V. and D. P. Ellis, “Audio Fingerprinting to Identify Multiple Videos of an Event”, *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2386–2389, IEEE, 2010.
3. Ojala, J., S. Mate, I. D. D. Curcio, A. Lehtiniemi and K. Väänänen-Vainio-Mattila, “Automated Creation of Mobile Video Remixes: User Trial in Three Event Contexts”, *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, MUM '14, pp. 170–179, ACM, New York, NY, USA, 2014.
4. McCormick, D., *Boston Marathon: Can Technology Do a Better Job of Finding Bombs?*, 2013, [http://spectrum.ieee.org/tech-talk/at-work/test-and-measurement/sniffing-out-explosives/?utm\\_source=techalert&utm\\_medium=email&utm\\_campaign=041813](http://spectrum.ieee.org/tech-talk/at-work/test-and-measurement/sniffing-out-explosives/?utm_source=techalert&utm_medium=email&utm_campaign=041813), [Accessed April 2013].
5. Anil Alexander, D. T., Oscar Forth, “Music and Noise Fingerprinting and Reference Cancellation Applied to Forensic Audio Enhancement”, *Audio Engineering Society Conference: 46th International Conference: Audio Forensics*, 2012.
6. Dixon, S. and G. Widmer, “MATCH: A Music Alignment Tool Chest.”, *Music Information Retrieval (ISMIR), 6th International Conference on*, pp. 492–497, 2005.
7. Müller, M., H. Mattes and F. Kurth, “An Efficient Multiscale Approach to Audio Synchronization.”, *Music Information Retrieval (ISMIR), 7th International Conference on*, pp. 192–197, 2006.

8. Montecchio, N. and A. Cont, “A Unified Approach to Real Time Audio-to-Score and Audio-to-Audio Alignment Using Sequential Montecarlo Inference Techniques”, *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 193–196, IEEE, 2011.
9. Joder, C., S. Essid and G. Richard, “Learning Optimal Features for Polyphonic Audio-to-Score Alignment”, *Audio, Speech, and Language Processing, IEEE Transactions on*, Vol. 21, No. 10, pp. 2118–2128, 2013.
10. Turetsky, R. J. and D. P. Ellis, “Ground-truth Transcriptions of Real Music from Force-Aligned Midi Syntheses”, *Music Information Retrieval (ISMIR), 4th International Conference on*, pp. 135–141, 2003.
11. Wang, A., “An Industrial Strength Audio Search Algorithm.”, *Music Information Retrieval (ISMIR), 4th International Conference on*, pp. 7–13, 2003.
12. Ikezoye, V. and J. Schrempp, *Method and Apparatus for Identifying Media Content Presented on A Media Playing Device*, 2004, <https://www.google.com/patents/US6834308>, US Patent 6,834,308.
13. Laroche, J., *Process for Identifying Audio Content*, 2002, <https://www.google.com/patents/US6453252>, US Patent 6,453,252.
14. Camarena-Ibarrola, A., E. Chávez and E. S. Tellez, “Robust Radio Broadcast Monitoring Using A Multi-band Spectral Entropy Signature”, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 587–594, Springer, 2009.
15. Haitsma, J. and T. Kalker, “A Highly Robust Audio Fingerprinting System.”, *Music Information Retrieval (ISMIR), 3rd International Conference on*, pp. 107–115, 2002.
16. Weinstein, E. and P. Moreno, “Music Identification With Weighted Finite-State

- Transducers”, *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, Vol. 2, pp. II-689, IEEE, 2007.
17. Dupraz, E. and G. Richard, “Robust frequency-based Audio Fingerprinting”, *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 281–284, March 2010.
  18. Fenet, S., G. Richard, Y. Grenier *et al.*, “A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting.”, *Music Information Retrieval (ISMIR), 12th International Conference on*, pp. 121–126, 2011.
  19. Müller, M., F. Kurth and M. Clausen, “Audio Matching via Chroma-Based Statistical Features.”, *Music Information Retrieval (ISMIR), 6th International Conference on*, pp. 288–295, 2005.
  20. Kurth, F. and M. Muller, “Efficient Index-Based Audio Matching”, *Audio, Speech, and Language Processing, IEEE Transactions on*, Vol. 16, No. 2, pp. 382–395, 2008.
  21. Anguera, X., A. Garzon and T. Adamek, “MASK: Robust Local Features for Audio Fingerprinting”, *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pp. 455–460, IEEE, 2012.
  22. Ke, Y., D. Hoiem and R. Sukthankar, “Computer Vision for Music Identification”, *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, pp. 597–604, IEEE, 2005.
  23. Baluja, S. and M. Covell, “Audio Fingerprinting: Combining Computer Vision & Data Stream Processing”, *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, Vol. 2, pp. II-213, IEEE, 2007.
  24. Viola, P. and M. Jones, “Rapid Object Detection Using A Boosted Cascade of Simple Features”, *Computer Vision and Pattern Recognition, 2001. CVPR 2001.*

- Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, pp. I–511, IEEE, 2001.
25. Cano, P., E. Battle, T. Kalker and J. Haitsma, “A Review of Audio Fingerprinting”, *Journal of VLSI signal processing systems for signal, image and video technology*, Vol. 41, No. 3, pp. 271–284, 2005.
  26. Weber, J. L. and E. W. Myers, “Human Whole-Genome Shotgun Sequencing”, *Genome Research*, Vol. 7, No. 5, pp. 401–409, 1997.
  27. Brown, M. and D. G. Lowe, “Automatic Panoramic Image Stitching Using Invariant Features”, *International Journal of Computer Vision*, Vol. 74, No. 1, pp. 59–73, 2007.
  28. Shrestha, P., M. Barbieri and H. Weda, “Synchronization of Multi-camera Video Recordings Based on Audio”, *Proceedings of the 15th international conference on Multimedia*, pp. 545–548, ACM, 2007.
  29. Bryan, N. J., P. Smaragdis and G. J. Mysore, “Clustering and Synchronizing Multi-camera Video via Landmark Cross-Correlation”, *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 2389–2392, IEEE, 2012.
  30. Shrestha, P., M. Barbieri, H. Weda and D. Sekulovski, “Synchronization of Multiple Camera Videos Using Audio-Visual Features”, *Multimedia, IEEE Transactions on*, Vol. 12, No. 1, pp. 79–92, Jan 2010.
  31. Shrestha, P., H. Weda, M. Barbieri, E. H. Aarts *et al.*, “Automatic Mashup Generation from Multiple-camera Concert Recordings”, *Proceedings of the international conference on Multimedia*, pp. 541–550, ACM, 2010.
  32. Moorer, J., *Acoustic Pattern Identification Using Spectral Characteristics to Synchronize Audio and/or Video*, 2014, <https://www.google.com/patents/>

- US8849432, US Patent 8,849,432.
33. Bello, J. P., L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler, “A Tutorial on Onset Detection in Music Signals”, *Speech and Audio Processing, IEEE Transactions on*, Vol. 13, No. 5, pp. 1035–1047, 2005.
  34. Guggenberger, M., “Multimodal Alignment of Videos”, *Proceedings of the ACM International Conference on Multimedia*, MM '14, pp. 667–670, ACM, New York, NY, USA, 2014.
  35. Llagostera Casanovas, A. and A. Cavallaro, “Audio-Visual Events for Multi-camera Synchronization”, *Multimedia Tools and Applications*, Vol. 74, No. 4, pp. 1317–1340, 2015.
  36. Bishop, C., *Pattern Recognition and Machine Learning*, Vol. 1, Springer New York, 2006.
  37. Forney Jr, G., “Generalized Minimum Distance Decoding”, *Information Theory, IEEE Transactions on*, Vol. 12, No. 2, pp. 125–131, 1966.
  38. Basaran, D., A. Cemgil and E. Anarim, “Model Tabanlı Ses Dizisi Hizalanması (Model Based Audio Sequence Alignment)”, *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, pp. 606–609, 2011, (in Turkish).
  39. Basaran, D., A. T. Cemgil and E. Anarim, “Model Based Multiple Audio Sequence Alignment”, *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, pp. 13–16, IEEE, 2011.
  40. Basaran, D., A. Cemgil and E. Anarim, “A Probabilistic Model-Based Approach for Aligning Multiple Audio Sequences”, *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, Vol. 23, No. 7, pp. 1160–1171, July 2015.
  41. Basaran, D., A. Cemgil and E. Anarim, “SMC Samplers for Multiresolution Audio

- Sequence Alignment”, *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 201–205, May 2013.
42. Troughton, P. and S. Godsill, “Bayesian Model Selection for Time Series Using Markov Chain Monte Carlo”, *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, Vol. 5, pp. 3733–3736, Apr 1997.
43. Kashino, K. and S. Godsill, “Bayesian Estimation of Simultaneous Musical Notes Based on Frequency Domain Modelling”, *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, Vol. 4, pp. 3305–3308, May 2004.
44. Cemgil, A. T., C. Fevotte and S. J. Godsill, “Variational and Stochastic Inference for Bayesian Source Separation”, *Digital Signal Processing*, Vol. 17, No. 5, pp. 891–913, 2007, special Issue on Bayesian Source Separation.
45. Lee, S.-Y. and K. G. Lee, “Synchronous and Asynchronous Parallel Simulated Annealing with Multiple Markov Chains”, *Parallel and Distributed Systems, IEEE Transactions on*, Vol. 7, No. 10, pp. 993–1008, Oct 1996.
46. Basaran, D., *Demonstration of the Sequential Alignment Algorithm for Multiple Audio Sequences*, 2014, [http://www.dogacbasaran.com/Dogac\\_Basarans\\_home\\_page/MultisequenceAlignment.html](http://www.dogacbasaran.com/Dogac_Basarans_home_page/MultisequenceAlignment.html), May 2014.
47. Del Moral, P., A. Doucet and A. Jasra, “Sequential Monte Carlo Samplers”, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 68, No. 3, pp. 411–436, 2006.