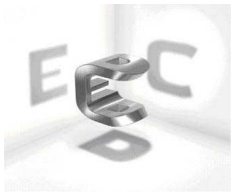
ECD Master Thesis



Mixed Supervision Latent Dirichlet Allocation for Automatic Bird Songs Identification System using Audio Recordings

Béatrice MOISSINAC

15/09/2011

Supervision: Raviv RAICH, Assitant Professor

Location: School of Electrical Engineering and Computer Science - Oregon State University - Corvallis,Oregon - USA

Abstract:

We aim to elaborate an automatic bird songs identification system using audio recordings. We use a mixed supervision LDA model to construct a probabilistic model able to learn the songs, and therefore predict the species on an instance-level.

Résumé :

Ce rapport condense 4 mois de travail sur l'élaboration d'un système automatique de reconnaissance d'espèce d'oiseaux utilisant des enregistrements audio. Nous utilisons le LDA, pour construire un modèle probabiliste supervisé capable d'apprendre la structure des chants d'oiseaux, et pouvoir ainsi prédire l'espèce.

Contents

1 Hosting Institution **2**

2 Acknowledgement **2**

3 Contribution of Authors **2**

4 About this Work **2**

5 Introduction **3**
5.1 Motivation and Background . . . 3
5.1.1 Representation of Bird Songs . . . 3
5.1.2 Automatic Identification System of Bird Species using Audio Recordings . . . 3
5.2 Bioacoustic Project . . . 4
5.2.1 Recording . . . 4
5.2.2 Segmentation . . . 5
5.2.3 Classification . . . 6
5.3 Organization of this Thesis . . . 6

6 Bioacoustics Literature Review **6**
6.1 Gaussian Mixture Model . . . 6
6.2 Hidden Markov Model . . . 7
6.3 Neural Network . . . 7
6.4 Dynamic Time Warping . . . 8
6.5 Sinusoidal Modeling . . . 8

7 Supervised Latent Dirichlet Allocation **9**
7.1 Introduction to LDA and Topic Modeling . . . 9
7.1.1 Parameters and Indices of LDA . . . 9
7.1.2 Parameters and Indices of sLDA . . . 10
7.2 Mixed Supervision Latent Dirichlet Allocation . . . 10
7.3 Problem Statement . . . 12
7.4 Learning MSLDA using Gibbs Sampling . . . 12
7.4.1 Sampling S . . . 13
7.4.2 Sampling θ . . . 15
7.4.3 Sampling ϕ . . . 17

8 Experiments with Synthetic Data Sets **18**
8.1 Generative Process for Proposed MSLDA . . . 18
8.2 Implementation Details . . . 18
8.3 Simulation Details . . . 19
8.4 Results . . . 19

9 Experiments with Data Sets from the Field **23**
9.1 Data Set Details . . . 23
9.2 Implementation Details . . . 25
9.3 Simulation Details . . . 25
9.4 Results . . . 25
9.5 Discussion on Results . . . 26

10 Conclusion **26**

A First appendix - Detailed results for section 8 **II**

B Second appendix - Apriori Algorithm Results X

List of Tables

1 List of parameters for MSLDA 12
2 Estimation of θ, mixture of song, for document 1 21
3 Extract from Table 7 and 9 21
4 Estimation of θ II
5 Γ matrix II
6 Estimation of ϕ III
7 Estimation of ϕ - Continuing IV
8 Estimation of ϕ - Continuing V
9 Estimation of ϕ - Continuing VI

List of Figures

1 Spectrogram of a recording 4
2 Hand-labeled spectrogram 5
3 Automatic segmentation 5
4 (a): Graphical model for LDA (b): Graphical model for sLDA 9
5 Graphical model for Automatic bird species Identification System 11
6 Subset of generated documents 19
7 true ϕ of Corpus 20
8 Estimated ϕ over 1000 iterations averaged every 100 iterations 20
9 Sampling of generated documents with $\alpha = 1$ and $\beta = 0.02$ 22
10 Sampling of generated documents with $\alpha = 2$ and $\beta = 0.01$ 23
11 Example of homogeneous cluster 24
12 Example of heterogeneous cluster 24
13 (a): Number of occurrences for syllable (1) among bird species (b):Number of occurrences for syllable (16) among bird species 25
14 (a): Number of occurrence for each song of species (1) (b): Number of occurrence for each song of species (7) 26
15 Sampled ϕ at iteration 500 VII
16 Sampled ϕ at iteration 600 VII
17 Sampled ϕ at iteration 700 VIII
18 Sampled ϕ at iteration 800 VIII
19 Sampled ϕ at iteration 900 IX
20 Sampled ϕ at iteration 1000 IX

List of Algorithms

1 Learning MSLDA using Gibbs Sampling 14
2 Generative Process for proposed MSLDA 18

1 Hosting Institution

School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon, United States of America.

2 Acknowledgement

First, I would like to express my sincere gratitude to my internship advisor, Dr. Raviv RAICH, for welcoming me into his research team. Dr. RAICH gave me the opportunity to learn how to lead a research project, guided me into the right direction each time it was needed, and taught me that everything should be written in less than 10 lines of Matlab code. I will emulate those qualities in the future. Furthermore, he gave the opportunity to come at OSU and meet the requirement for the Graduate Program in OSU. Therefore, I am indefinite thankful for this internship, which now allows me to emulate those qualities at OSU.

I am very thankful to my French advisor Dr. Julien VELCIN, the secretary of the Graduate Office Jennifer CHANTELOUP and Laura HAMPTON for signing all my paperwork. They have been very supportive in my journey to come and stay at Oregon State University, and it wouldn't have been possible without their patience toward my administrative -and legendary- bad luck. Living on the edge needs a lot of patience.

I would like to thank my officemates William BRENDEL, Mohamed AMER and Ali TORKAMANI for their useful advises in many fields, scientific or not. I have a specially thank for my research group, Behrouz, Gaole, and Greg for all the fun.

Thank to JA for hosting me and my passport. Thank to Vita and Habibi, for all the fun. Living on the edge demands good friends! :)

I am extremely grateful to my parents for the financial support and patience they gave to my project. More particularly, I am indefinitely indebted to Madame la Baronne Sidonie Echaubard de Lusclade.

3 Contribution of Authors

The Mixed Supervision Latent Dirichlet Allocation (MSLDA) model is a specialized application of LDA model to bird species identification. This development was developed in collaboration with Dr. Raviv RAICH, and is the continuation of previous modeling developed in [28, 27]. Forrest BRIGGS provided the instance-level data, the spectrogram and the segmented syllable images as developed in [5].

4 About this Work

This work is the technical report of my 4-months internship, in partial fulfillment of the requirements for the degree of Master 2 ECD.

5 Introduction

The birds are an undeniable part of the Earth's ecosystem. Their role as prey and predator for many other species, and their relatively easiness to detect make them an excellent target for environment monitoring [38]. In the current trend of protecting biodiversity, being able to collect and analyze data sufficiently and efficiently appears as an indispensable step. However, the collection of data by people can be biased by the observer or other environmental and ecological factors but also by its very high cost. In order to reduce this cost, automatic recording systems have been introduced.

Automatic recording systems allow acoustic sampling over a large temporal and spatial scales. However it raises new challenges: The recorder's surroundings may be noisy (e.g. stream, wind, human activities). Moreover, birds may sing simultaneously or birds may be far from the microphone. Many studies overcame this problem by collecting sounds from birds in cages or by targeting them with handheld directional microphones [11, 43, 44], In either case, a great deal of human intervention is involved. In this project, we have collected data directly in the forest, using song meters. The human intervention has been limited to the installation and the collect of the data once the hard drives were full.

The classical approach of an Automatic Identification System (AIS) uses the Fast Fourier Transform (FFT) to convert the recordings into spectrograms. These spectrograms make the bird songs "visible" for interpretation and therefore open to analysis. The next step of the AIS is the computation of features, also called segmentation. A lot of work has been done by several research groups, one of each, the Bioacoustics group at OSU has developed its own segmentation procedure in [5].

During this internship, we have been attempting to create an automatic bird species identification system, such as given an audio recording; predict the species for each call, and the set of all species heard in each recording. We propose an approach using the topic modeling theory Latent Dirichlet Allocation and probabilistic Bayesian network.

5.1 Motivation and Background

The birds are an important part of our ecosystem, and monitoring them is a crucial step to understand and protecting biodiversity. It addresses ecological and ornithological questions:

- What are the variations in bird populations, with respect to environmental and ecological factors?
- How does a bird species interact with other bird species through vocalization?
- What is the impact of human activities among different bird populations and behaviors?

So far, ornithological studies have been done using manual labor to collect data, which led to significant bias from the observers or other environmental and ecological factors. Reducing this bias and the prohibitive cost of data collection is the current challenge of automatic recognition systems. Such systems could enable us to follow bird populations through time period and different location.

5.1.1 Representation of Bird Songs

The audio recordings are classically represented by spectrograms computed using Fast Fourier Transform (FFT). A typical spectrogram from our recordings will present the frequencies of bird songs over time. We define a *phrase* as a long sequence of distinct *syllables* as presented on Figure 1. A syllable is a short structured sound emitted by a bird. A song is composed of one or several phrases, each of them presenting a distinctive structure, which we attempt to capture in this work.

5.1.2 Automatic Identification System of Bird Species using Audio Recordings

A classical Automatic Identification System (AIS) of bird species using audio recordings can be decomposed in several steps:

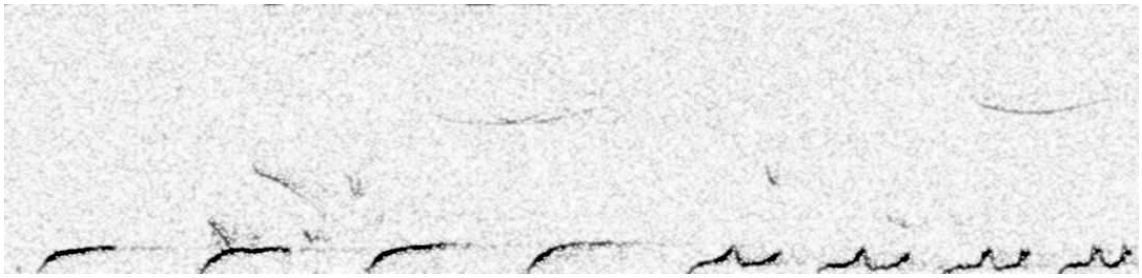


Figure 1: Spectrogram of a recording

1. Training Stage
   (a) Sample audio recording and labeling of the captured data files
   (b) Computation of spectrograms
   (c) Segmentation: identify component in a spectrogram that corresponds to syllables.
   (d) Featurization
   (e) **Classification of syllables**
2. Test Stage
   (a) Taking an unlabeled audio recording
   (b) Computation of spectrograms
   (c) Segmentation: identify component in a spectrogram that corresponds to syllables.
   (d) Featurization
   (e) **Classification of syllables**
   (f) Prediction of bird species labels

In this work, we have focused on the classification part (bolded lines) of the AIS developed within the Bioacoustics group.

5.2 Bioacoustic Project

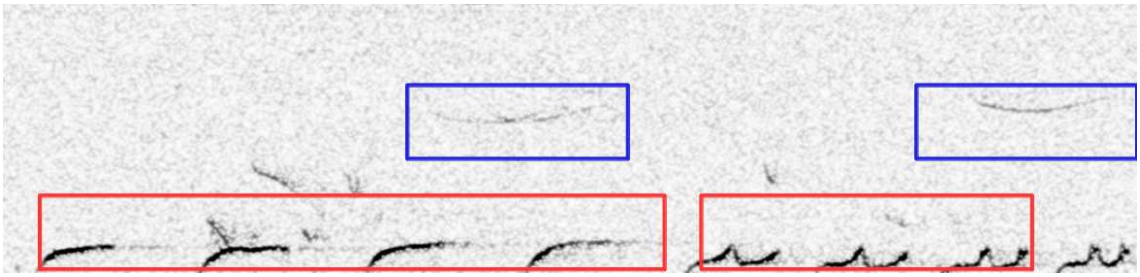
The Bioacoustics group[1] is the result of collaboration between the Forest Wildlife Landscape Ecology Department and the Electrical Engineering and Computer Science Department at Oregon State University. Researchers from ecology and machine learning backgrounds cooperate to develop sustainable data collection systems and algorithms to automatically identify bird species on audio recordings.

5.2.1 Recording

In collaboration with the Forest Wildlife Landscape Ecology Department, the Bioacoustics group has placed 13 Wildlife Acoustics Song Meters (SM1) in different locations of the H.J. Andrews experimental forest (Oregon, USA). Two omni-directional microphones are enclosed inside a wind shield, with a battery powered computer using 32 Gb flash-memory to store the data. The recording sessions took place during summer 2009, 2010 and 2011. The summer presents many advantages: there is almost no rainfall, a lot of birds are present, people are available for hiking in remote places to check the song meters. The human intervention has been limited to the installation and the collection of the data once the hard drives were full. More than one Terabyte of data has been collected, and only a very small part has been labeled by trained bird specialists due to its prohibitive cost. The recordings contain simultaneous bird calls and

[1] http://eecs.oregonstate.edu/research/bioacoustics/

Figure 2: Hand-labeled spectrogram



natural noises from the surrounding environment (e.g. stream, rain fall and wind).

In previous publications, the Bioacoustics group used a classical representation of audio recordings using spectrogram generated by FFT. In Figure 2, the recording has been labeled by a member of the Forest Wildlife Landscape Ecology Department, with boxes drawn by the labeler around the syllables identifying the species.

5.2.2 Segmentation

Figure 3: Automatic segmentation

The extraction of features from a recording is called *segmentation*. The quality of the classification is dependent on the quality of the syllable features, and therefore the quality of the segmentation of those features from the recordings. It is very important to note that labeling and segmentation are different processes. Segmentation is an automatic process extracting the syllables and their features from the spectrogram. On the other hand, labeling is a manual process and the boxes drew by the labeler are NOT used to segment the syllables.

During this internship, we have been using segmented data from the most recent segmentation process developed by the Bioacoutics group [5]. Shortly, we describe the segmentation process as follows:

1. Using a ten-second recording sampled at 16 kHz, we preprocess it and reduce the noise by normalizing the spectrogram and applying iteration of whitening filter.

2. Then, we apply a two-dimensional segmentation over time and frequency, which separates songs that could have been overlapping in the time dimension (but not in the frequency dimension). We used a SISL classifier to predict segmented masks on spectrograms.

3. Finally, we compute the features of each segment by cropping the mask from the spectrogram.

We have been using 38 features such as minimum frequency, maximum frequency, bandwidth, duration, area, perimeter, rectangularity. The detailed features and their analysis can be found in [28] and [5].

5.2.3 Classification

In [27], each syllable was characterized as a probability distribution and treated the feature representation of each frame encompassing a syllable to be observations from that particular syllable distribution. The Independent Frame Independent Syllable (IFIS) model and the Markov Chain Frame Independent syllable (MCFIS) models were introduced . In [6], we built a probabilistic model fed with audio features extracted from short intervals of time. Most recently, in [5], we used a multi-label multi-instance framework.

In [28], Lakshminarayanan presented an inference algorithm for a supervised Latent Dirichlet Allocation (sLDA). In this work, we will continue and develop this algorithm, to develop more efficient inference techniques.

5.3 Organization of this Thesis

In section 6, we will thoroughly review the related works on classification algorithms for automatic bird species identification systems. In section 7, we will introduce the Latent Dirichlet Allocation model and the Topic Modeling theory. We will also developed our own inference algorithm adapted to bird species identification. In section 8, we experiment with a synthetic set of data, used as sample to test our approach. In section 9, we experiment with real data sets from the field. Finally in section 10, we analyze and discuss the results before approaching future work.

6 Bioacoustics Literature Review

In this section, we present an extensive review of related works on classification algorithms for automatic bird species recognition using audio recordings. We divided this section according to the different methods used to classify the syllables. Some publications were using several methods, therefore, their results are presented several times using various methodologies.

6.1 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a well-known algorithm which has been widely used in speech processing. It is a probabilistic model which uses a combination of several multivariate Gaussian densities to model the distributions of the data. In other words, each bird species represents a different set of modeled frequency distributions.

In 2004, Kwan et al. published a first article using GMM and Hidden Markov Model (HMM) for automatic bird species recognition using bird songs [26]. The project aimed to minimize the number of bird strikes with planes around airports. They compared HMM and GMM algorithms and they have shown that GMM algorithms tend to achieve slightly better performance and a more suitable real-time computation.

In a more recent publication [25], the authors supported their choice of using GMM with two arguments. First, the multivariate Gaussian densities should be able to model underlying classes. Second, a linear combination of Gaussian is flexible enough to represent a large range of distribution of classes. Therefore, the goal was to find the Maximum A Posteriori (MAP) of a probability given the data. However, the authors assumed the distribution of birds equally likely (stating that information about birds is easily obtainable in airports), which is a very strong hypothesis that we will relax later using the Dirichlet distribution.

Another comparison of GMM and HMM but also Dynamic Time Warping (DTW), is published by Sumervuo et al. [41] in 2006. TLDR, They have shown that DTW presents the best accuracy among those models, when the parametrization is obtained by Mel-frequency cepstral coefficients (MFCC) based syllable trajectory .
Recently, Jancovic et al. [21] published a GMM model of automatic recognition of tonal bird sounds in noisy environments.

6.2 Hidden Markov Model

Hidden Markov Model is a popular model for speech recognition due to its ability to integrate different level of statistical language (distribution, features, .).

Kogan et al. were among the first to publish a method using HMM [24]. They also proposed a Dynamic Time Warping system, to be compared with HMM. However, they manually labelled the segments of syllables to apply the standard HMM.

In 2004, Kwan et al. [26] compared HMM and GMM. HMM is implemented with the Baum-Welch method, also known as Expectation Maximization (EM) as described in [35] They have shown that GMM algorithms tend to achieve better performance and a more suitable real-time computation. Alternatively, Sumervuo et al. [41] implement HMM using Viterbi training as an approximation of Baum-Welch [22]. Moreover, they considered the labels only on a song level, and not on a syllable or element level. Those studies [26, 41] have presented comparisons between GMM and HMM, and even if GMM tends to be slightly more accurate than HMM, this difference remains small.

The Hidden Markov Model is also used in [4],[43] and [12] and shows less performance in noisy environments.

6.3 Neural Network

The Neural Networks have been one of the earliest and most used methods in automatic bird song recognition. McIliraith and Card are pioneers in applying Neural Network to data base including a large number of bird species. In 1995, they published a first article [29], proposing a back-propagation neural network. They were able to demonstrate very early on, the critical importance of features used to feed the network. In 1997, they published three articles comparing different versions of neural networks for automatic identification of bird songs. In [32] and [30], McIlraith and Card used a back-propagation in two-layers Perceptron, and compared it to statistical methods like quadratic discriminant analysis. In [31], they proposed to use the short-time spectrum of the signal as a feature, and then apply a feed-forward neural network with back-propagation. They also considered a parametrization which accounts for the duration of silences, in addition of the elements and the songs. However, those models require a considerable amount of computation.

In 2004, David Chesmore [11] published his own recognition system: Intelligent Bioacousticsignal Identification System (IBIS) based on Time Domain Signal Coding (TDSC) and Artificial Neural Network (ANN). The results seem extremely encouraging but it requires a good quality of sound. In our LDA-based system, the noisy environment is an important part of our implementation. The same year, Chesmore and Ohya [10] used also this method to identify four British grasshoppers. They obtained accuracies between 70-100% depending on the sound quality and background noise, which is quite on average to what the other methods obtained.

In 2005, Selouani et al. [40] published a method that combined Time Delay Neural Networks (TDNN) and an autoregressive version of a back-propagation network. This structure can handle the classification of syllables, as well as capturing their temporal structures. This system achieved encouraging performance compared to the basic back-propagation-based neural network. Their technique improved the NN

methods by implementing a feedback loop to the multilayer perceptron.

In [7], the authors proposed to use "past" and "future" frames as well as current frame as inputs to the neural network, in order to introduce a dynamic process. In [23], Juang et al.used a prediction-based singleton-type recurrent neural fuzzy networks In [36], the authors developed a method to compute distances between syllables, and then, create a self-organizing neural network.

6.4 Dynamic Time Warping

Dynamic Time Warping (DTW) [37] is used to align and compare sequences with varying lengths. Therefore, DTW is a popular algorithm to compare syllables of different duration even if it is computationally very expensive and may include background information that is not relevant for identification or that may be less performant in noisy environments.

The earliest attempt occurred in 1996, when Anderson et al. [1] constructed a DTW recognition system using template matching of signal spectrograms. However, their system needed manual segmentation of the syllables.

In [24], Kogan et al. introduced a system of bird song identification based on matching the spectrogram using templates, which are the reference sequences for the classes. However, their method requires a person to first select the spectrogram templates by hand, and therefore this is an important limitation to apply this method on larger amount of varying data. Moreover, the algorithm was not working very well, when used with a noisy environments or short duration vocalizations.

In [41], Somervuo et al. uses DTW in a modeling recognition system, and then compare it to sinusoidal modeling via GMM and HMM. They have shown that DTW presents the best accuracy among those models when the parametrization is obtained by Mel-frequency cepstral coefficients (MFCC) based syllable trajectory.

6.5 Sinusoidal Modeling

The sinusoidal modeling is based on the representation of a set of time-varying sinusoidal components. Once the features are extracted, classical classification algorithms can be applied, like the nearest neighbor [41].

In [17], the authors build an automatic recognition system for fourteen European birds. They used a sinusoidal modeling of syllables to extract features from the sounds. Even if this is a very simplified model (each syllable is characterized by frequency and amplitude trajectory of only one time-varying sinusoid), the results were significant. The same authors, developed this model in [18], by introducing four parameters which represent the harmonic structure of the syllable. This model is revised in [41] and is compared to DTW, GMM and HMM. They have shown that sinusoidal modeling has a high level of accuracy for species whose syllables mostly belong to a certain class of harmonicity. However, this modelization was too simple for other types of bird sounds.

Fagerlund published in 2004 his thesis about automatic recognition of bird species by their sounds [13]. This thesis is an extensive study about sinusoidal modeling developed by Härmä.

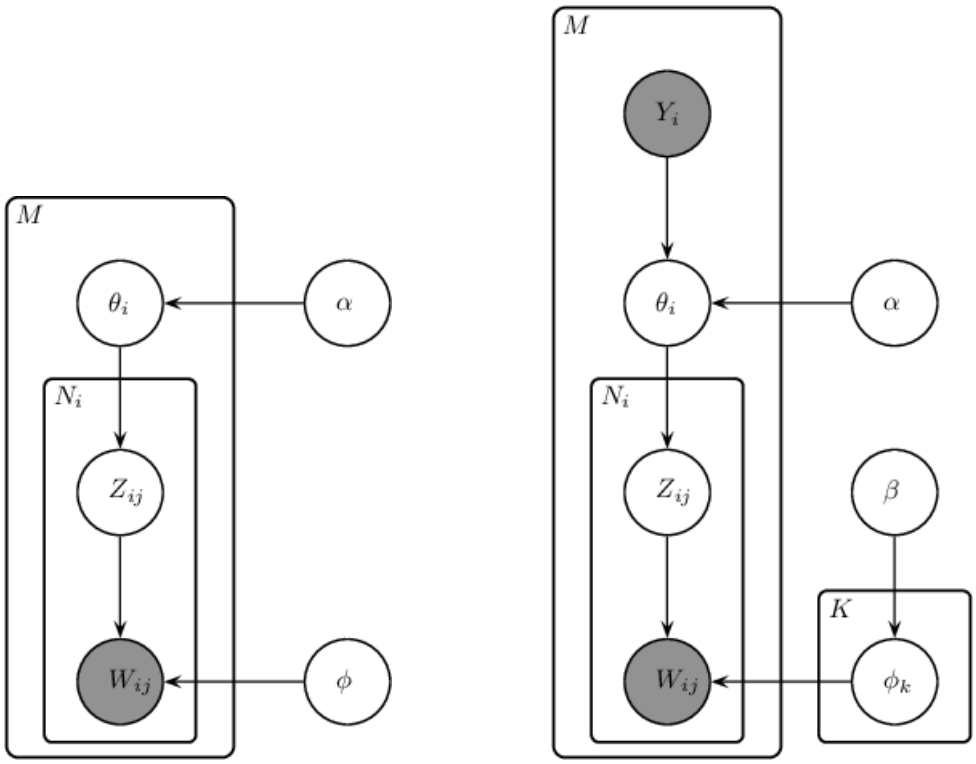
In 2006, Chen and Maher [9] developed a technique that calculate the degree to which the derived parameters match a set of stored templates that were determined from a set of reference bird vocalizations. The use of spectral peak tracks is simple and robust to noise. Moreover, it has a relatively low computational complexity. The authors have shown that this method performed better than DTW or HMM.

Alternatively, in [39] the authors developed a method handling the inharmonic bird sounds using wavelet coefficient feeding a self organizing map and a multilayer perceptron.

7 Supervised Latent Dirichlet Allocation

7.1 Introduction to LDA and Topic Modeling

The Latent Dirichlet Allocation (LDA) is introduced by Blei, Ng and Jordan in [3]. It is originally an unsupervised method for topic modeling, that is, a method to uncover topic mixtures using words in a text corpus of document. Each document forms a *bag-of-words* containing words as instances. Each bag-of-word is independent from the others. This ability to decompose a topic into a mixture of words gives a great flexibility to topic modeling.
Latent Dirichlet Allocation presents several advantages, among which, it enables us to represent a model via a hierarchical graphical model. The Figure 4(a) shows the graphical model for unsupervised LDA. Each node is a variable: the white nodes are said to be *latent variables*, such as the variables that they represent are unknown. The known variables are represented with a shaded node.



(a) (b)

Figure 4: (a): Graphical model for LDA (b): Graphical model for sLDA

7.1.1 Parameters and Indices of LDA

Parameters

Mixture of topics is the distribution of proportion for each topic within one document. We write it as θ_i, the $K \times 1$ vector of proportion for document i such as $\theta_i \sim Dirichlet(\alpha)$.

Topics are the classes of each word. We write it as Z_{ij}, for the topic of the j^{th} word in the i^{th} document.

Words are the instances contained in the documents. We write W_{ij} as the j^{th} word in the i^{th} document.

Parameters α and β characterize respectively the distribution of θ and W

Indices

Number of documents : $i = 1 \dots M$

Number of words : $j = 1 \dots N_i$ and total number of words in the corpus $N = \sum_{i=1}^{M} N_i$.

7.1.2 Parameters and Indices of sLDA

The supervised LDA, in [14], is an ingenious continuation of the classical LDA model. Supervised LDA enables the utilization of additional information on the bag-level included in Y. Figure 4(b) shows its graphical model.

The main difference states in the drawing of class labels Y_i for each document. This means that the Dirichlet prior of θ is a $K \times M$ matrix of M vectors. Each vector is computed such as $Y_i \times \alpha$, in order for the i^{th} column of the Dirichlet prior to correspond to the i^{th} column of Y, the matrix of classes.

Parameters

Classes is the bag-level label, which indicates for each document, which topic is present or not using a disjunctive matrix. We write it as Y_i, the vector of classes in the i^{th} document. Y is a $C \times M$ matrix of classes.

Mixture of topics is the distribution of proportion for each topic within one document. We write it θ_i, the $K \times 1$ vector of proportion for document i such as $\theta_i \sim Dirichlet(Y_i \times \alpha)$.

Topics are the classes of each word. We write it as Z_{ij} for the topic of the j^{th} word in the i^{th} document.

Words are the instances contained in documents. We write W_{ij} as the j^{th} word in the i^{th} document.

Mixture of words is the distribution of proportion for each word within one topic. We write is ϕ_i the $V \times 1$ vector of proportion for topic k such as $\phi_k \sim Dirichlet(\beta)$.

Parameters α and β characterize respectively the distribution of θ and ϕ

Indices

Number of classes : $c = 1 \dots C$

Number of documents : $i = 1 \dots M$

Number of words : $j = 1 \dots N_i$ and total number of word in the text corpus $N = \sum_{i=1}^{M} N_i$.

Number of topics : $k = 1 \dots K$

7.2 Mixed Supervision Latent Dirichlet Allocation

The topic modeling theory in bioacoustic analysis requires us to adapt the vocabulary we are using. The *documents* are the audio recording of bird songs, the *words* will be the syllables present on the spectrogram. As in human speech, each occurrence of a syllable is unique, but can be classified in a cluster with its center as a generalized representation of this syllable. We can therefore generate a vocabulary of syllables (a codebook) that can be learned by the model.

In a preliminary approach, we were working directly with a model such as in Figure 4(b), but the results obtained were difficult to interpret. The difference between the classes C and the topics K was very ambiguous. To resolve this ambiguity, we introduced the parameter S, the bird songs, such as each audio recording has a mixture of songs, and each song has a mixture of syllables. Therefore, the parameter S is very much symmetric to the topic parameter in the general sLDA model.

Moreover, we transfered the bird species parameter Z to be dependent from S. It is also possible that the syllable has not been labeled. In this case, Z is said to be unknown. Several arguments led us to suppose that the distribution of the songs are different if Z is known or unknown:

- The labeler may not label a bird that is difficult to identify for various reasons (noise, ambiguity, inability of the labeler)
- The labeler may not label every occurrence of one syllable, if this one is very dominant in the recording. (e.g. "5 times is enough, the machine will figure out the rest of it")

We know that "easy" birds are more often very well labeled, and that the "difficult" birds are labeled less often or not at all. Therefore, we can assume that the distribution of songs are different if Z is known or unknown. This differentiation appears with the two parameters S^1 and S^2. S^1 is the song parameter when Z is known, S^2 is the parameter when Z is unknown. Figure 5 presents this graphical model, which continues the work done in [28]. Since we do not estimate the unknown Z's during the sampling, they are not present in the second part of the graph.

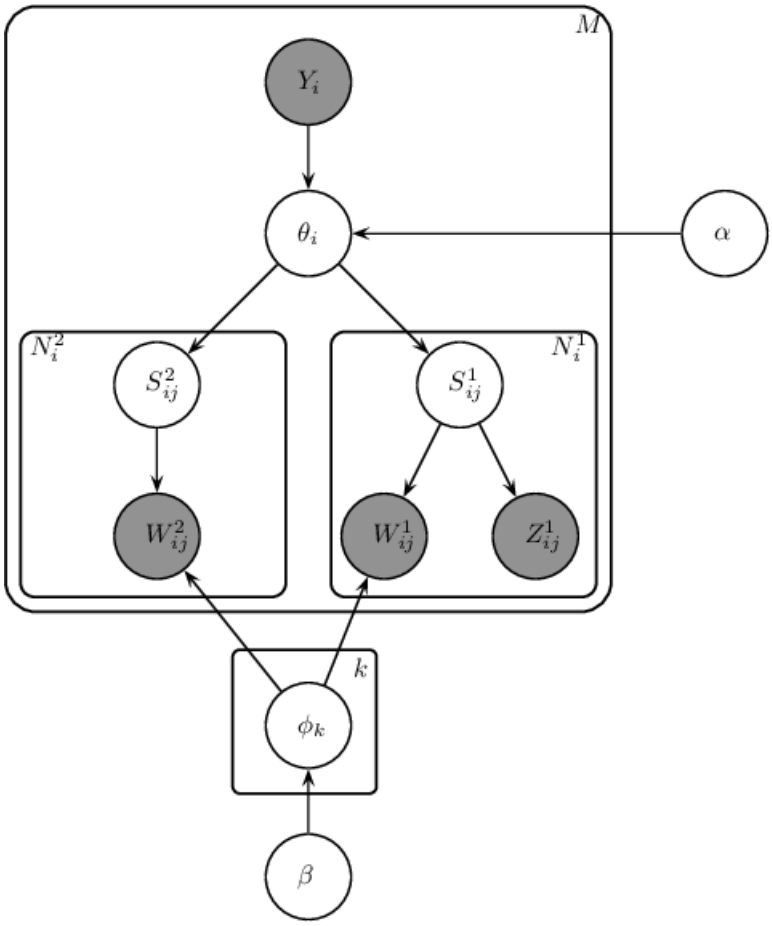


Figure 5: Graphical model for Automatic bird species Identification System

Parameters

Classes are the bag-level labels, which indicates each recording in which species are present or not using a binary matrix. We write it as Y_i, the vector of classes in the i^{th} recording. Y is a $C \times M$ matrix of classes.

Mixture of songs is the distribution of proportion for each song within one audio recording. We write it as θ_i, the $K \times 1$ vector of proportion for recording i such as $\theta_i \sim Dirichlet(Y_i \times \alpha)$. The probability of the k^{th} song given the i^{th} audio recording is given by θ_{ki}.

Bird species are the instance-level labels on each syllable. We write it Z_{ij} for the species of the j^{th} word in the i^{th} audio recording.

Syllables are the instances contained in documents. We write W_{ij} as the j^{th} syllable in the i^{th} audio recording.

Mixture of syllable is the distribution of proportion for each syllable cluster within one song. We write is ϕ_k the $V \times 1$ vector of proportion for song k such as $\phi_k \sim Dirichlet(\beta)$. The probability of the j^{th} syllable given the k^{th} song is given by ϕ_{jk}.

Parameters α and β characterize respectively the distributions of θ and ϕ.

Indices

Number of species : $c = 1 \dots C$, is an index defining the classes.

Number of syllables : $j = 1 \dots N_i$ and total number of syllables in the corpus $N = \sum_{i=1}^{M} N_i$. This index defines as the words in a traditional topic model.

Number of clustered syllables : $v = 1 \dots V$ cluster. This index must be understood as the vocabulary of a dictionary that our model is learning.

Number of songs : $k = 1 \dots K$, this index defines topics.

Number of recordings : $i = 1 \dots M$, this index defines documents.

Table 1: List of parameters for MSLDA

M	Number of recordings
N_i	Number of syllables in recording i
V	Vocabulary size
C	Number of species
K	Number of songs
S_i	$N_i \times 1$ vector of song assignments corresponding to syllables in recording i
W_i	$N_i \times 1$ vector of all the syllables in recording i
Z_i	$N_i \times 1$ vector of species assignment corresponding to syllables in recording i
Y	$C \times M$ presence/absence matrix of a species in a recording
θ	$K \times M$ matrix whose the i^{th} column is the mixture of songs for recording i
ϕ	$V \times K$ matrix whose the k^{th} column is the mixture of words for the song k
α	Part of Dirichlet prior of θ
β	Dirichlet prior of ϕ

7.3 Problem Statement

Using the model developed in section 7.2 we will attempt to solve the following problem: learn a model for (W, S, Z) such as the classification of a new recording i is possible with minimum errors.

7.4 Learning MSLDA using Gibbs Sampling

In [2], the authors compared different methods using LDA modeling. They implemented various inference solutions: Variational Bayes (VB), collapsed Gibbs sampling (CGS), collapsed variational Bayes (CVB) and also maximum a-posteriori (MAP). We already studied the effect of those inference algorithms for supervised LDA in [28]. We showed that with proper hyperparameters (α and β), CVB and VB tend to have the same performance, however MAP is significantly worse even if it is computationally more efficient.

The MSLDA model introduced in this work (fig. 5) is more complex than in [28] (fig. 4(b)). This model requires a more complex inference due to the additional latent variables. Therefore, we used a Gibb Sampling (GS) method to learn the latent variables within our model.

Gibbs Sampling is a Markov Chain Monte Carlo (MCMC) algorithm, and more specifically, GS is a special case of the Metropolis-Hastings algorithm. The method generates random variables from a marginalized distribution without calculating the density. A convergence proof can be found in section 3 of [8], even when monitoring this converge is not easy. Griffiths et al. have been the first to derive a collapsed Gibbs Sampling (ϕ and θ are marginalized out, therefore α and β are calculated after the sampling) in [16]. Other examples can be found in [15, 34, 19]. An awesome and **complete** derivation of collapsed GS has been independently published by Wang in [45].

In this work, we did not use a collapsed Gibbs Sampling, because we wanted to be able to use additional information contained in ϕ and θ (e.g. the bag-level and instance-level label) during the sampling. Therefore, we have derived an inference solution to learn LDA using GS. The algorithm 1 gives the detailed procedure.

7.4.1 Sampling S

In section 7.2, we have explained the differences between the distribution of known and unknown labeled instances. In this section, we will derive the full conditional distributions of S^1 and S^2.

Full conditional distribution of S when Z is known: The sampling of S^1 with GS requires us to compute the full conditional distribution of S^1, that is, calculate the distribution of S^1 given its Markov blanket. In Figure 5, the Markov Blanket of S^1_{ij} is the set of variables W^1_{ij}, Z^1_{ij} and θ_i. In equation (1), ϕ_k is also present, because of its influence on W^1_{ij}, but then ϕ_k is canceled out with the summation to S^1.

$$P(S^1_{ij} = k|W^1_{ij} = v, Z^1_{ij} = c, \theta_i, \phi_k) = \frac{P(S^1_{ij}, Z^1_{ij}, W^1_{ij}, \theta_i, \phi_k)}{P(Z^1_{ij}, W^1_{ij}, \theta_i, \phi_k)} \tag{1}$$

$$= \frac{P(W^1_{ij} = v|S^1_{ij} = k, \phi_k)P(Z^1_{ij}|S^1_{ij})P(S^1_{ij}|\theta_i)}{P(W^1_{ij}, Z^1_{ij}, \theta_i, \phi_k)} \tag{2}$$

$$= \frac{\phi_{vk}P(Z^1_{ij} = c|S^1_{ij} = k)\theta_i(k)}{\sum_K \phi_{vk}P(Z^1_{ij} = c|S^1_{ij} = k)\theta_i(k)} \tag{3}$$

$\sum_K \phi_{vk}P(Z^1_{ij} = c|S^1_{ij} = k)\theta_i(k) = constant$ with respect of K. Those probabilities will sum to 1 with respect of S^1's.

We will now present the derivation of equation (3), which is composed by three probabilities:

- Conditional distribution of S^1 given θ
- Conditional distribution of W^1 given S^1
- Conditional distribution of Z^1 given S^1

Conditional distribution of S: $P(S^1_{ij} = k|\theta_i)$ The probability for the song of the j^{th} syllable in the i^{th} recording to be the k^{th} song, given the mixture of songs θ_i in the i^{th} recording is $\theta_{k,i}$, the value at the k^{th} line and i^{th} column of the matrix θ. See section 7.4.2 for the sampling of θ.

Conditional distribution of W: $P(W^1_{ij} = v|S^1_{ij} = k, \phi_k)$ The probability for the j^{th} syllable in the i^{th} recording to be the v^{th} word in the vocabulary, given the k^{th} song and its mixture of syllable ϕ_k is $\phi_{v,k}$, the value at the v^{th} line and k^{th} column of the matrix ϕ. See section 7.4.3 for the sampling of ϕ.

Algorithm 1 Learning MSLDA using Gibbs Sampling

Initialisation
Set all counts to zero
for $i = 1$ to M **do** {Each document}
  for $j = 1$ to N_i **do** {Each word}
    Sample song index $S_{ij} = k \sim Mult(1/K)$
    Increment count song-recording $\Omega(k, i) = \Omega(k, i) + 1$
    Increment count word-song $\Psi(W_{ij} = v, k) = \Psi(W_{ij} = v, k) + 1$
  end for
end for
Construct Γ, $K \times C$ binary matrix since one song is used only by one species
Estimate α {See section 6.4.1}
for $i = 1$ to M **do** {Each document}
  Sample $\theta_i \sim Dirichlet(Y_i \times \alpha + \Omega(:, i))$
end for
for $k = 1$ to K **do** {Each song}
  Sample $\phi_k \sim Dirichlet(\beta + \Psi(:, k))$
end for

Gibbs Sampling
for $iter = 1$ to $iterMax$ **do**
  Set the counts $\Omega(k, i)$ and $\Psi(v, k)$ to zero.
  for $i = 1$ to M **do** {Each document}
    for $j = 1$ to N_i **do** {Each word}
      Resample S using equation (3)
      Increment count song-recording $\Omega(k, i) = \Omega(k, i) + 1$
      Increment count word-song $\Psi(W_{ij} = v, k) = \Psi(W_{ij} = v, k) + 1$
    end for
  end for
  for $i = 1$ to M **do** {Each document}
    Sample $\theta_i \sim Dirichlet(Y_i \times \alpha + \Omega(:, i))$
  end for
  for $k = 1$ to K **do** {Each song}
    Sample $\phi_k \sim Dirichlet(\beta + \Psi(:, k))$
  end for
  After burning period and every x iteration:
  $\theta_{est} = \theta_{est} + \theta$ {Add current θ estimate to the global estimate.}
  $\phi_{est} = \phi_{est} + \phi$ {Add current ϕ estimate to the global estimate.}
  count = count + 1;
end for
Average the estimates:
$\theta_{est} = \frac{\theta_{est}}{count}$
$\phi_{est} = \frac{\phi_{est}}{count}$

Conditional distribution of Z: $P(Z_{ij}^1 = c|S_{ij}^1 = k)$ The probability for the species corresponding to the j^{th} syllable in the i^{th} recording to be the c^{th} species, given the k^{th} song is a deterministic relationship:

$$P(Z_{ij}^1 = c|S_{ij}^1 = k) = \begin{cases} 1 & \text{if the song belongs to this species} \\ 0 & \text{otherwise} \end{cases}$$

This corresponds to line 10 in algorithm 1. We have constructed an arbitrary $K \times C$ binary matrix, crossing each song with each species such that each song belongs to only one species. A species can have as many songs as needed. Meaning that in equation 3, S_{ij}^1 will be sampled only in the range that belongs to Z_{ij}^1.

Full conditional distribution of S when Z is unknown: The sampling of S^2 with GS requires to compute the full conditional distribution of S^2, that is, calculate the distribution of S^2 given its Markov blanket. In Figure 5, the Markov Blanket of S_{ij}^2 is the set of variables W_{ij}^2 and θ_i. Similarly to the derivation of S^1, we compute:

$$P(S_{ij}^2 = k|W_{ij}^2 = v, \theta_i, \phi_k) = \frac{P(S_{ij}^2, W_{ij}^2, \theta_i, \phi_k)}{W_{ij}^2, \theta_i, \phi_k)} \tag{4}$$

$$= \frac{P(W_{ij}^2 = v|S_{ij}^2 = k, \phi_k)P(S_{ij}^2|\theta_i)P(\theta_i)}{P(W_{ij}^2, \theta_i, \phi_k)} \tag{5}$$

$$= \frac{\phi_{vk}\theta_i(k)}{\sum_K \phi_{vk}\theta_i(k)} \tag{6}$$

$\sum_K \phi_{vk}\theta_i(k) = constant$ with respect of K. Those probabilities will sum to 1 with respect of S^2's.

Since Z_{ij}^2 is unknown, the probability of any species matching to any song is equal to one. No restriction is imposed to the range of song.

We will now present the derivation of equation (3), which is composed by two probabilities:

- Conditional distribution of S^2 given θ
- Conditional distribution of W^2 given S^2

Conditional distribution of S: $P(S_{ij}^2 = k|\theta_i)$ The probability for the song of the j^{th} syllable in the i^{th} recording to be the k^{th} song, given the mixture of songs θ_i in the i^{th} recording is $\theta_{k,i}$, the value at the k^{th} line and i^{th} column of the matrix θ. See section 7.4.2 for the sampling of θ.

Conditional distribution of W: $P(W_{ij}^2 = v|S_{ij}^2 = k, \phi_k)$ The probability for the j^{th} syllable in the i^{th} recording to be the v^{th} word in the vocabulary, given the k^{th} song and its mixture of syllable ϕ_k is $\phi_{v,k}$, the value at the v^{th} line and k^{th} column of the matrix ϕ. See section 7.4.3 for the sampling of ϕ.

7.4.2 Sampling θ

The sampling of θ with GS requires us to compute the full conditional distribution of θ, that is, calculate the distribution of θ given its Markov blanket. As present in Figure 5, the Markov Blanket of θ_i is the set of variables $\{S_{i1}^1 ... S_{iN}^1\}, \{S_{i1}^2 ... S_{iN}^2\}, Y_i$ and α.

$$P(\theta_i|\{S_{i1}^1...S_{iN}^1\},\{S_{i1}^2...S_{iN}^2\},Y_i,\alpha) = \prod_j^N P(S_{ij}^1|\theta_i)f(\theta_i|Y_i,\alpha) \tag{7}$$

$$= \prod_{k=1}^K \theta_{k,i}^{\Omega(k,i)+Y_i\times\alpha} \tag{8}$$

$$\tag{9}$$

$$\theta_i \sim Dirichlet(\Omega(k,i) + Y_i \times \alpha) \tag{10}$$

The matrix $\Omega(k,i)$, in which we report the number of times the song k was assigned inside recording i, for each syllable.

$$\Omega(k,i) = \sum_{i=1}^{M}\sum_{j=1}^{N_i} I(s_{ij} = k \wedge m = i) \tag{11}$$

The transition from equation (7) to (8) and then (10) is done by using the Dirichlet distribution properties. Those "gritty details" are available in [45].

We will now present the derivation of equation (8), which is composed by two probabilities:

- Conditional distribution of S_i given θ
- Density function of θ_i given Y_i and α

Conditional distribution of S_i: $P(\{S_{i1}...S_{iN}\}|\theta_i)$ We multiply the i^{th} column of the θ matrix, therefore, there is no distinction between S^1 and S^2 anymore.

$$\{S_{i1}...S_{iN}\}|\theta_i \sim Discrete(\theta_i)$$

$$P(\{S_{i1}...S_{iN}\}|\theta_i) = \prod_{j=1}^N P(S_{ij}|\theta_i) \tag{12}$$

$$= \prod_{k=1}^K \theta_{k,i}^{\Omega(k,i)} \tag{13}$$

Conditional density of θ: $f(\theta_i|Y_i,\alpha)$

$$\theta \sim Dirichlet(Y_i \times \alpha)$$

This means that the Dirichlet prior of θ is a $K \times M$ matrix. Each column vector is computed as $Y_i \times \alpha$, in order for the i^{th} column of the Dirichlet prior to correspond to the i^{th} column of Y, the matrix of classes.

$$f(\theta|Y_i \times \alpha) = \frac{\prod_K \theta_{i,k}^{Y_i\times\alpha-1}}{B(Y_i \times \alpha)} \tag{14}$$

About the estimation of α We did not satisfyingly estimate α YET. This is an important part to the future works for this project. The estimation of α can be done using the Polya distribution, also known as the Dirichlet-multinomial distribution. In preliminary researches, we had implemented such estimations using a gradient descent, but the algorithm was not converging fast enough. Therefore we postponed the improvement of this optimization. The complete derivation is presented by Tom MINKA in [33]. A study of several optimization algorithms to estimate α can be found in [20].

7.4.3 Sampling ϕ

The sampling of ϕ with GS requires us to compute the full conditional distribution of ϕ, that is, calculate the distribution of ϕ given its Markov blanket. As present in fig. 5, the Markov Blanket of ϕ is the set of variables $\{W_{i1}^1...W_{iN}^1\}, \{W_{i1}^2...W_{iN}^2\}$ and β.

$$P(\phi_k|\{W_{i1}^1...W_{iN}^1\}, \{W_{i1}^2...W_{iN}^2\}, \beta) = constant \times \prod_j^N P(W_{ij}^1|S_{ij}^1, \phi_k) f(\phi_k|\beta) \tag{15}$$

$$= \prod_V \phi_{v,k}^{\Psi(v,k)+\beta} \tag{16}$$

$$\tag{17}$$

$$\phi_k \sim Dirichlet(\Psi(.,k) + \beta) \tag{18}$$

The matrix $\Psi(v,k)$, in which we report the number of times the song k was assigned to syllable cluster v across the corpus.

$$\Psi(v,k) = \sum_{j=1}^{N} I(W_{ij} = v \wedge S_{ij} = k) \tag{19}$$

The transitions from equation (15) to (16) and then (18) are done by using the Dirichlet distribution properties. Those "gritty details" are available in [45].

We will now present the derivation of equation (8), which is composed by two probabilities:

- Conditional distribution of W_i given $\{S_{i1}...S_{iN}\}$ and ϕ_k
- Density function of ϕ_k given β

Conditional distribution of W_i: $P(\{W_{i1}...W_{iN}\}|\{S_{i1}...S_{iN}\}, \phi_k)$ We multiply the k^{th} column of the ϕ matrix, therefore, there is no distinction between W^1 and W^2.

$$\{W_{i1}...W_{iN}\}|\{S_{i1}...S_{iN}\}, \phi_k \sim Discrete(\phi_k)$$

$$P(\{W_{i1}...W_{iN}\}|\{S_{i1}...S_{iN}\}, \phi_k) = \prod_{j=1}^{N} P(W_{ij}|S_{ij}) \tag{20}$$

$$= \prod_{i:v=1}^{V} \prod_{k=1}^{K} \phi_{v,k}^{\Psi(v,k)} \tag{21}$$

Conditional density of ϕ: $f(\phi_k|\beta)$

$$\phi_k \sim Dirichlet(\beta)$$

$$P(\phi|\beta) = \prod_{k=1}^{K} P(\phi_k|\beta) \tag{22}$$

$$= \prod_{k=1}^{K} \frac{1}{B(\beta)} \prod_{v=1}^{V} \phi_{k,v}^{\beta-1} \tag{23}$$

About β: Similarly to α, it is possible and necessary to estimate β.The estimation of β can be done using the Polya distribution, also known as the Dirichlet-multinomial distribution. The complete derivation is presented by Tom MINKA in [33]. A study of several optimization algorithms to estimate α can be found in [20].

8 Experiments with Synthetic Data Sets

In the first experiment, we generate a synthetic set of data - a toy model - to test our approach.

8.1 Generative Process for Proposed MSLDA

We generated a random set of data using the standard generative process for MSLDA [3]. We adapted it to integrate our new parameters. The algorithm 2 gives the details of the procedure.

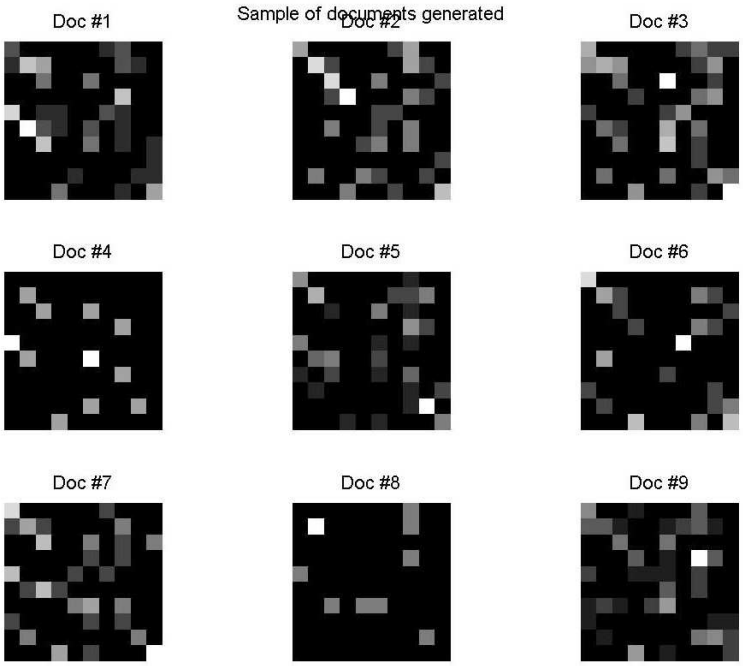
Algorithm 2 Generative Process for proposed MSLDA

Construct Γ, $K \times C$ binary matrix such as one song is used only by one species
for $i = 1$ to M **do** {Each document}
  Randomly generate the number of word N_i
end for
for $k = 1$ to K **do** {Each song}
  Sample True_$\phi_k \sim Dirichlet(\beta)$)
end for
Randomly generate Y
Randomly generate α
for $i = 1$ to M **do** {Each document}
  Sample True_$\theta_i \sim Dirichlet(Y_i \times \alpha)$
  for $j = 1$ to N_i **do** {Each word}
    Sample $S_{ij} = k \sim Discrete(\theta_i)$
    Sample $W_{ij} = v \sim Discrete(\phi_k)$
    Use Γ to find the corresponding Z_{ij}
  end for
end for

8.2 Implementation Details

The algorithms 1 and 2 have been implemented entirely in Matlab. We used vectorization to make the algorithms more efficient with Matlab since Matlab does not handle loops well. The sampling from the

Figure 6: Subset of generated documents



Dirichlet distribution has been done using the awesome FastFit toolbox developed by Tom MINKA[2]. This toolbox requires the use of Lightspeed toolbox, also developed by Tom MINKA[3].

8.3 Simulation Details

We generate several models with various numbers of recordings of bird species, song, song per species, α and β parameters. The following simulation details have been chosen for presentation because it shows a clear and understandable output. It is important to understand how we may interpret the results at this stage, thus the reader may confidently comprehend section 9 using the real data.

We generate a model with 250 recordings, 4 bird species, 4 song per species (so 16 songs in total) and 100 types of syllable for a little more than 12,000 instances. α and β have been fixed respectively to 2 and 0.02, so we have sparse mixtures. We run the Gibbs Sampling algorithm during 1,000 iterations with a burn-in of 500 iterations. We saved the estimations of θ and ϕ every 100 iteration after this burn-in.

8.4 Results

Figure 6 is a subset of the first nine documents generated. Those matrices are normalized probability matrices. Each cell corresponds to the normalized probability of one word appearing in the document. The shade of grey indicates the level of probability. The syllable cluster having the highest probability will have a white cell. The syllable-cluster having the smallest probability will have a black cell. These matrices are used in the same way as any other figure presented in this work. It is important to notice that those matrices are filled up first by column, that is, $v = 1$ will be in $(1, 1)$, and $v = 2$ will be in $(2, 1)$ (second line, first column).

First, we generate the documents using algorithm 2 and obtain 250 documents as in Figure (6).We also generate the true ϕ (Figure (7)) in order to compare them with the ϕ that we will estimate later. We run algorithm 1 for the generated documents. The ϕ generated during the GS at iterations 500, 600,

[2]http://research.microsoft.com/en-us/um/people/minka/software/fastfit/
[3]http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/

Figure 7: true ϕ of Corpus

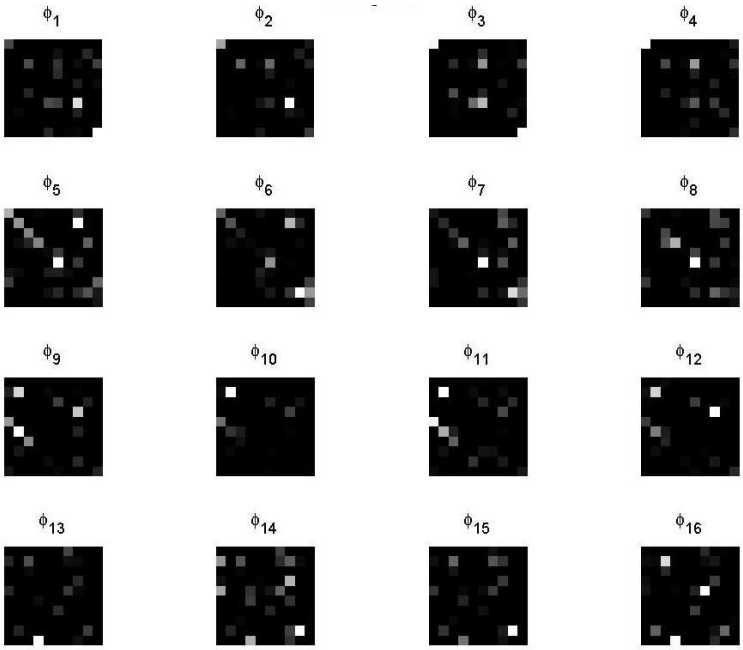


Figure 8: Estimated ϕ over 1000 iterations averaged every 100 iterations

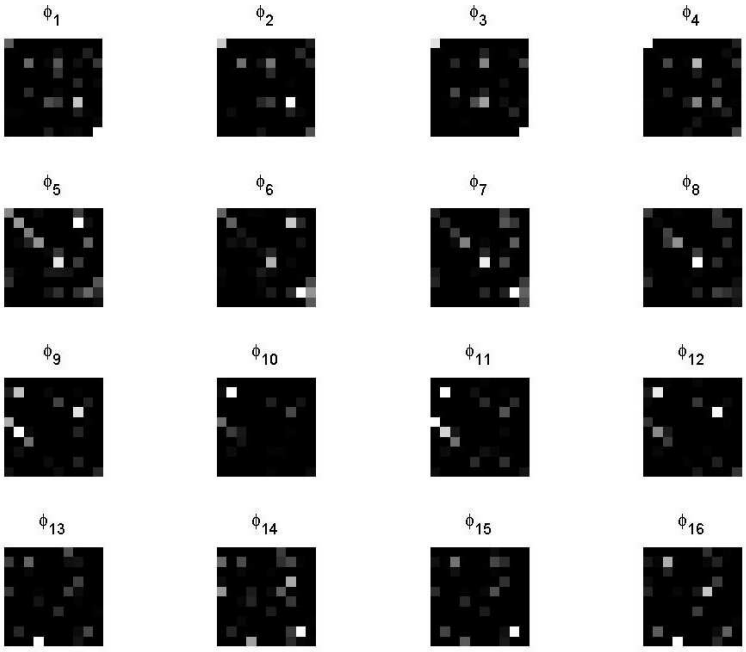


Table 2: Estimation of θ, mixture of song, for document 1

	Doc 1
θ_1	0.0095
θ_2	0.0195
θ_3	**0.1609**
θ_4	0.0180
θ_5	0.0000
θ_6	0.1046
θ_7	0.0000
θ_8	0.0001
θ_9	**0.3391**
θ_{10}	0.0000
θ_{11}	0.0905
θ_{12}	0.0000
θ_{13}	**0.2121**
θ_{14}	0.0458
θ_{15}	0.0000
θ_{16}	0.0000

Table 3: Extract from Table 7 and 9

W_{ij}	$P(W_{ij}\|S_{ij}) = 9$
5	0.1303
12	0.1476
16	0.1934
27	0.0853
74	0.1717

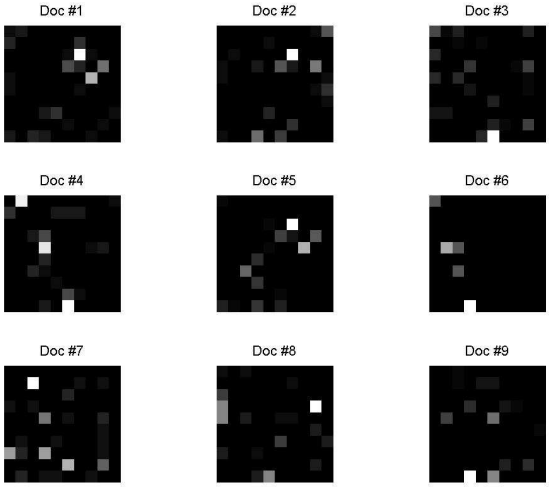
700, 800, 900, and 1000 (Figures 15,16,17,18,19 and 20 in Appendix A) are added up and divided by 5 to obtain the estimated ϕ (Figure 8).

The matrices of estimated ϕ (Figure 8) are obtained by running the GS (algorithm 1), adding the estimated ϕ every 100 iterations after the burn-in, and averaging. Those matrices should have a structure that correspond to the true ϕ (Figure (7)). Simply speaking, the "structure" made by the white cells should be similar. The matrices may shuffle within the same bird species, that is, it may happen that the matrix true ϕ_1 would be at position of estimate ϕ_4, but still within its species range (e.g. 1 to 4 for species 1).

From θ to ϕ, how to navigate in LDA output?: We will explore document 1 with the purpose of exposing the process of analysis of MSLDA results. The first step is to look at the estimated θ, the mixture of song for the document 1. The complete table of θ estimates are shown is Table (4) of the Appendix A. We presented an extract in Table 2. It's important to notice, that those probabilities sum to 1, which is expected from a Dirichlet distribution.

The highest probabilities are $\theta_{(9,1)} = 0.3391$,$\theta_{(13,1)} = 0.2121$ and $\theta_{(3,1)} = 0.1609$. This means that the songs 9,13 and 3 are most likely to appear in this document. We also know from matrix Γ that those songs respectively correspond to species 3, 4 and 1 (Table 5 of Appendix A). We will now have a look at the estimates of ϕ. Unlike θ, which can be a relatively small matrix (at least in this example), ϕ is a

Figure 9: Sampling of generated documents with $\alpha = 1$ and $\beta = 0.02$



100×16 matrix. The complete matrix ϕ is presented in Tables 6,7, 8 and 9 in Appendix 1. We recall that ϕ is a $V \times S$ matrix of mixture of syllable per song. Therefore, each column is a mixture of syllables summing to 1 for each song. There are a lot of zeros in both matrices, but the reader should not be mistaken. The probability is just too small to be written significantly to the then thousands place. A probability will never be equal to zero, but will be extremely small.

By reading the columns 9, 13 and 3 in Tables (6,7,8,9), we are able to reconstruct the likeliest syllables among the likeliest songs within document 1. We have done this for song 9 as an example to the reader in Table 3. The most common syllables of song 9 are syllables 5, 12, 16, 27 and 74. They are syllables with the highest probabilities in column 9 of the ϕ matrix. In Figure 8, matrix ϕ_9, those syllables are the one with the most white cells. Visually, it is easy to see that the "shape" made by the white squares in ϕ_9 can be found back in document 1 of Figure 6 (e.g. the diagonal made with 5,16 and 27).

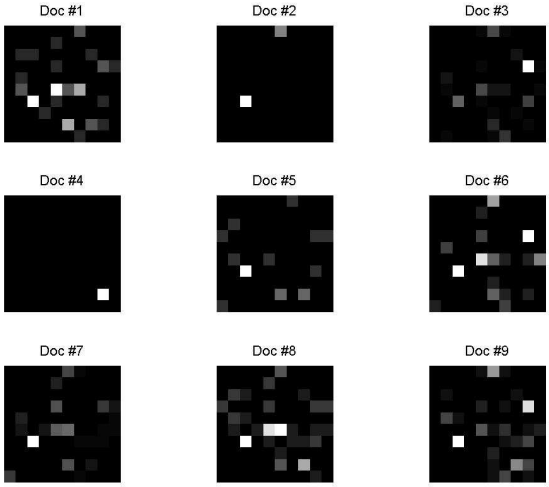
It is crucial to notice the repetition of "shapes" among songs of a same species. For instance, ϕ_5, ϕ_6, ϕ_7 and ϕ_8 have a distinctive diagonal on the upper left corner. This is consistent with the notion of repertoire for a bird: consistently repeating song behavior. It is the probability of each syllable and its proportion, that distinguishes song from other song of the same repertoire. The fact that we can observe such a structure confirms the relevance of this model.

On the effect of β and α In order to illustrate the effect of β and α on the model, we simulated two examples in figures 9 and 10. The figure 9 was generated with the same parameters explained in section 8.3, except with α set to 1. The documents generated clearly present a poorer mixture of songs than with $\alpha = 2$ in figure 6. More exactly, the number of songs present in the mixture of each document is decreased.

The figure 10 was generated with the same parameters as explained in section 8.3, except that β set to 0.01. The documents generated clearly present a poorer mixture of songs than with $\beta = 0.02$ in Figure 6. More exactly, the number of words present in the mixture of each song is decreased.

This demonstrates the crucial importance of estimating β and α in future work, because their effect on the model is a fundamental part of the inference and therefore, of the learning.

Figure 10: Sampling of generated documents with $\alpha = 2$ and $\beta = 0.01$



9 Experiments with Data Sets from the Field

9.1 Data Set Details

In section 5.2.1, the presentation of data collection methods were described. In section 5.2.2, it was explained how the syllables were segmented. After segmentation, a set of 548 recordings were obtained, each 10 seconds in length. This set contains 10,232 segmented syllables with 38 features characterizing minimum frequency, maximum frequency, bandwidth, duration, area, perimeter, rectangularity, etc... The detailed features and their analysis can be found in [28, 5]. The labeling has been done on the bag-level and instance-level with 13 bird species.

The repartition of bird species per recording is not uniform. We ran the Apriori algorithm to extract the frequent "birdsets" in our data. From the bag-level labeling, we construct the matrix Y (as seen in section 7.2) and use it as a transaction matrix for the Apriori algorithm. We used a Java Applet developed by the University of Regina[4] (Canada). The complete output is shown in Appendix B. It was not relevant to use the Support has a decision parameter, therefore we fixed it to a very low level (0.1).

We will comment on the results obtained with a higher degree of confidence. We obtain the following most frequent "birdsets": [1 2 3, 1 4 6, 2 3 13, 9 10 11], which means that the algorithm may be confused by the co-occurrence of syllables from birds of a same set. We have no way of preventing the algorithm from constructing a song composed of syllables from birds belonging to the same set. On the other hand, species (12) will always be the only species in the recording, so we can reasonably assume that it's identification will be much easier. Moreover, there are 4,998 instances where the species is known, and 5,234 instances with unknown labels causing us to use only the set of 4,998 data with known labels.

Features clustering The dimensional reduction of features into clusters has been done using a multi-class linear dimension reduction via a generalized Chernoff bound [42]. We used 15 dimensions and 100 clusters obtaining a vocabulary $V = 100$. In Figure (11), we present a very homogeneous cluster. Each syllable has its species indicated on the top and the composition of this cluster was mainly composed of species (2). There are three syllables from species (12], but since those syllables are extremely similar to the others, we can reasonably assume that the labeler made a mistake.

[4]http://www2.cs.uregina.ca/ dbd/cs831/notes/itemsets/itemsetgenerator.php

Figure 11: Example of homogeneous cluster

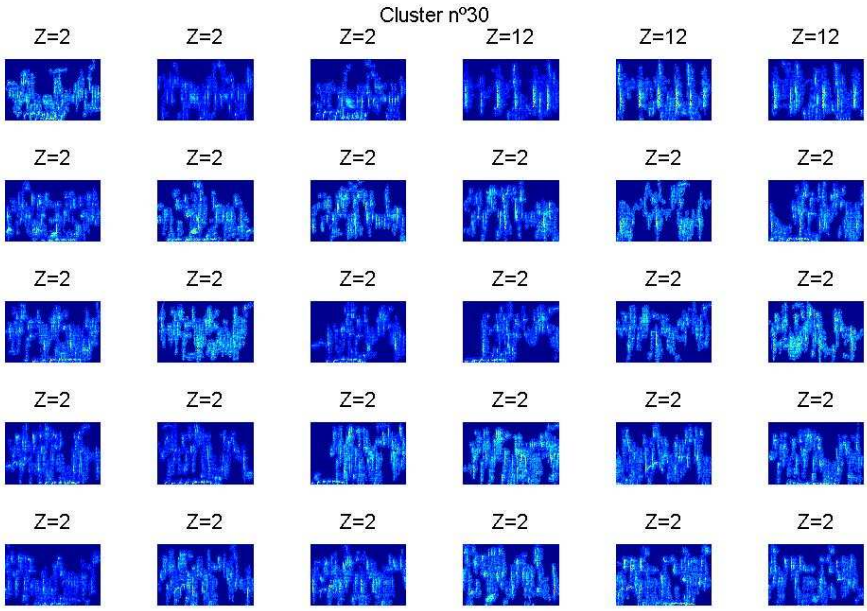
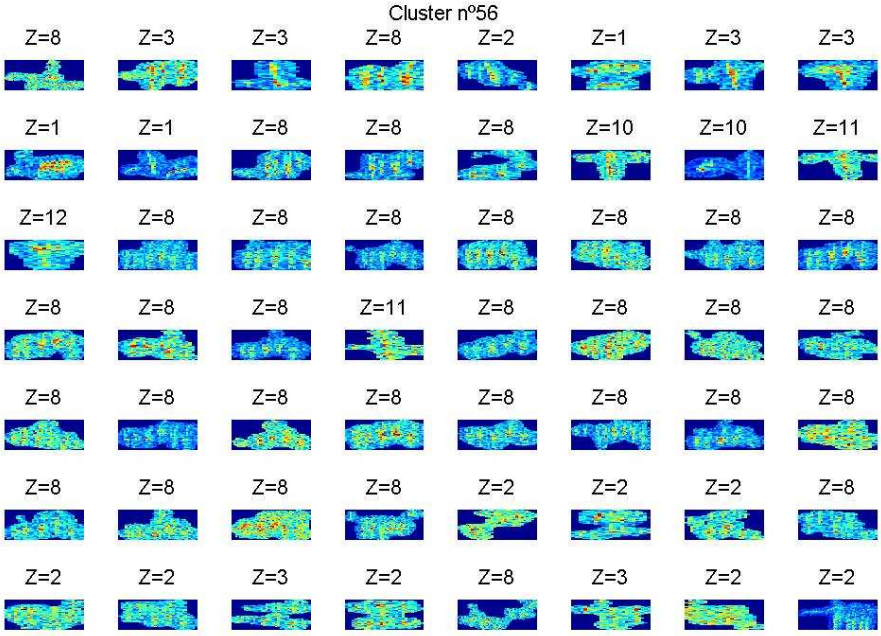
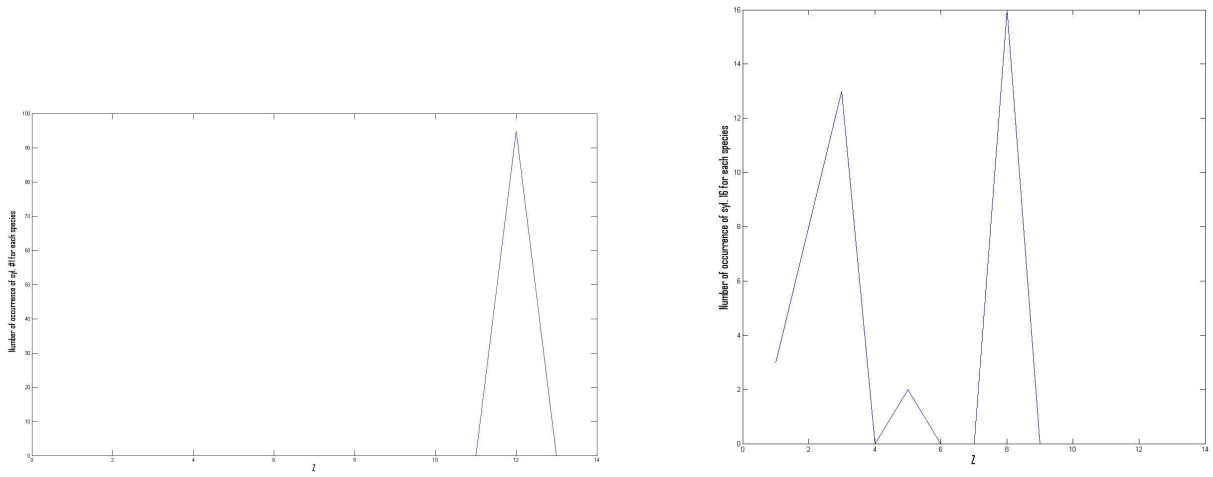


Figure 12: Example of heterogeneous cluster





(a) (b)

Figure 13: (a): Number of occurrences for syllable (1) among bird species (b):Number of occurrences for syllable (16) among bird species

On the other hand, Figure (12) produces a lesser homogeneous cluster. The syllables look noisier and the labeled species are too numerous to assume that they were mislabeled. We have judged this cluster of a hundred as being homogeneous enough to be used for learning. Nevertheless, this may be an area of approach for improvement in future work. As it has been demonstrated before, the quality of the clustering greatly determines the quality of the learning.

9.2 Implementation Details

The algorithm (1) has been implemented entirely in Matlab. We used vectorization to make the algorithms more efficient with Matlab since Matlab does not handle loops well. The sampling from the Dirichlet distribution has been done using the awesome FastFit toolbox developed by Tom MINKA[5]. This toolbox requires the use of Lightspeed toolbox, also developed by Tom MINKA[6].

9.3 Simulation Details

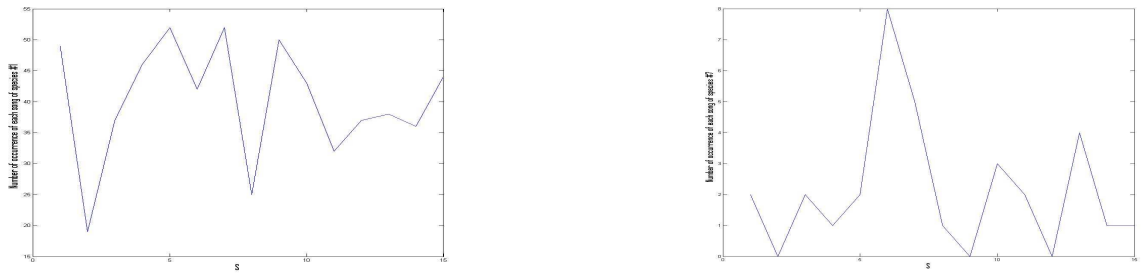
Training Stage We have run the Gibbs Sampling for 3,000 iterations, using a burn-in of 1,000 with the α values set to 2 and β values set to 0.02. After the burn in we added the estimated ϕ and θ every 100 iterations.

Test Stage We run the simulation again using 90%, 95% and 99% of the initial 4,998 labeled data set. Then, we use the matrix Γ and the sampled songs S to predict the species for the second part of the data set.

9.4 Results

Training Stage At the training stage, we used the data set with 4,998 known labeled instances to learn the model. After the sampling, we can construct a matrix recording the occurrences of each word by each species. In Figure 13(a), the reader can see that syllable (1) is characterized to only species (1). This is a very encouraging result, but not every syllable has this distinctive distribution. In Figure 13(b), the distribution of syllable (16) is much less accurately characterized to a particular species.

[5]http://research.microsoft.com/en-us/um/people/minka/software/fastfit/
[6]http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/



(a) (b)

Figure 14: (a): Number of occurrence for each song of species (1) (b): Number of occurrence for each song of species (7)

Similarly, we would expect the same result with the distribution of the songs belonging to one species. We would like to see among the range of songs for a species, two or three very particular songs in a species that seldomly occurs. In Figure 14(b), we clearly observe this result on song (7). Nevertheless, in Figure 14(a), the distribution of songs over this species is much more uniform.

Test Stage We obtain an error rate averaging around 85%. In section 9.5, we will discuss this problem and propose several ways to improve it.

9.5 Discussion on Results

The result from the prediction is not a surprise for several reasons:

- We did not estimate either the α or β values, even if as demonstrate in section 8.4, they have a predominant role in the modeling.
- We know that there are some mistakes in the labels.
- We limited the number of songs to 15, and as seen in figure 14(a). If we do not allocate enough space for the songs to spread, we won't see particular picks appearing.

Those problems will be subject to future work by the Bioacoustics group.

Experimentation with Clustering: We run another classification algorithm obtaining the clusters from the multi-class linear dimension reduction via a generalized Chernoff bound.

We made the cluster determine the most frequent species found within this particular cluster. We remind the reader that we did this clustering only with the data set containing 4,998 labels. Each clustered determined its species, therefore we could draw a deterministic relationship between a syllable and its cluster to predict which species was present. Taking again the split of 90%, 95% and 99% of our datasets, we hid the labels for the rest of the dataset. Therefore a syllable without a label will take the label that was determined by the cluster.

Using this simplistic method, we obtain an accuracy of 75% when comparing the real label and the predicted label! This demonstrates that our clustering has a satisfying quality, and should not be the source of the problem.

10 Conclusion

During this internship, we have been attempting to create an automatic bird species identification system, such as given an audio recording; predict the species for each call, and the set of all species heard in each recording. We developed an application of LDA called Mixed Supervision LDA. Using both the

information from the bag-level and the instance-level, we constructed an inference to estimate this model. The first results were not matching our expectation. Therefore, there are several axes of amelioration on which the Bioacoustics group will put its efforts. To increase the accuracy, we will estimate α and β and insure the purity of our labeling. Also, we will study the effort of the number of syllable clusters and the number of song per species.

Automatic recording systems allow acoustic sampling over a large temporal and spatial scales. Therefore, the ability to analyze sufficiently and efficiently those data can be a very important step into the monitoring of biodiversity.

References

[1] S. E Anderson, A. S Dave, and D. Margoliash. Template-based automatic recognition of birdsong syllables from continuous recordings. *Journal of the Acoustical Society of America*, 100(2):1209–1219, 1996.

[2] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009.

[3] D. M Blei, A. Y Ng, and M. I Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] T. S Brandes. Feature vector selection and use with hidden markov models to identify frequency-modulated bioacoustic signals amidst noise. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(6):1173–1180, 2008.

[5] F. Briggs, B. Lakshminarayanan, L. Neal, X. Fern, R. Raich, M.G. Betts, S. Frey, and A. Hadley. Acoustic classification of multiplie simultaneous bird species: a multi-instance multi-label approach. *submitted to Journal of the American Statistical Association*.

[6] F. Briggs, R. Raich, and X. Z Fern. Audio classification of bird species: a statistical manifold approach. In *2009 Ninth IEEE International Conference on Data Mining*, pages 51–60, 2009.

[7] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang. Sensor network for the monitoring of ecosystem: Bird species recognition. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 293–298, 2007.

[8] George Casella and Edward I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992. ArticleType: research-article / Full publication date: Aug., 1992 / Copyright Â© 1992 American Statistical Association.

[9] Z. Chen and R. C Maher. Semi-automatic classification of bird vocalizations using spectral peak tracks. *J. Acoust. Soc. Am*, 120:5, 2006.

[10] D. Chesmore. Automated bioacoustic identification of species. *Anais da Academia Brasileira de CiÃªncias*, 76(2):436–440, 2004.

[11] E. D. Chesmore and E. Ohya. Automated identification of field-recorded songs of four british grasshoppers using bioacoustic signal recognition. *Bulletin of entomological research*, 94(04):319–330, 2004.

[12] W. Chu and D. T Blumstein. NOISE ROBUST BIRD SONG DETECTION USING SYLLABLE PATTERN-BASED HIDDEN MARKOV MODELS. *Training*, 1644:457, 2009.

[13] S. Fagerlund. *Automatic recognition of bird species by their sounds*. PhD thesis, Helsinki University of technology, 2004.

[14] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. Ieee.

[15] T. L Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.

[16] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. 2002.

[17] A. Harma. Automatic identification of bird species based on sinusoidal modeling of syllables. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V–545, 2003.

[18] A. Harma and P. Somervuo. Classification of the harmonic structure in bird vocalization. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 5, pages V–701, 2004.

[19] G. Heinrich. Parameter estimation for text analysis. *Web: http://www. arbylon. net/publications/text-est. pdf*, 2005.

[20] J. Huang. Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique Report*, 2005.

[21] P. Jancovic and M. Kokuer. Automatic detection and recognition of tonal bird sounds in noisy environments. 2011.

[22] B. H Juang and L. R Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(9):1639–1641, 1990.

[23] C. F Juang and T. M Chen. Birdsong recognition using prediction-based recurrent neural fuzzy networks. *Neurocomputing*, 71(1-3):121–130, 2007.

[24] J. A Kogan and D. Margoliash. Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study. 1998.

[25] C. Kwan, K. C. Ho, G. Mei, Y. Li, Z. Ren, R. Xu, Y. Zhang, D. Lao, M. Stevenson, V. Stanford, et al. An automated acoustic systemto monitor and classify birds. *EURASIP journal on applied signal processing*, 2006:52–52, 2006.

[26] C. Kwan, G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, and K.C. Ho. Bird classification algorithms: theory and experimental results. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V–289–92, Montreal, Que., Canada, 2004.

[27] B. Lakshminarayanan, R. Raich, and X. Fern. A Syllable-Level probabilistic framework for bird species identification. In *2009 International Conference on Machine Learning and Applications*, pages 53–59, 2009.

[28] Balaji Lakshminarayanan. *Probabilistic models for classification of bioacoustic data*. Master's thesis, Oregon State University, 2010.

[29] A. L McIlraith and H. C. Card. Birdsong recognition with DSP and neural networks. In *WESCANEX 95. Communications, Power, and Computing. Conference Proceedings. IEEE*, volume 2, pages 409–414, 1995.

[30] A. L. McIlraith and H. C. Card. Bird song identification using artificial neural networks and statistical analysis. In *Electrical and Computer Engineering, 1997. IEEE 1997 Canadian Conference on*, volume 1, pages 63–66, 1997.

[31] A. L McIlraith and H. C Card. Birdsong recognition using backpropagation and multivariate statistics. *Signal Processing, IEEE Transactions on*, 45(11):2740–2748, 1997.

[32] A. L. McIlraith and H. C. Card. A comparison of backpropagation and statistical classifiers for bird identification. In *Neural Networks, 1997., International Conference on*, volume 1, pages 100–104, 1997.

[33] T.P. Minka. Estimating a dirichlet distribution. *Annals of Physics*, 2000(8):1–13, 2003.

[34] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, 2008.

[35] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1 edition, April 1993.

[36] Louis Ranjard and Howard A. Ross. Unsupervised bird song syllable classification using evolving neural networks. *The Journal of the Acoustical Society of America*, 123(6):4358, 2008.

[37] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[38] C.H. Sekercioglu, G.C. Daily, and P.R. Ehrlich. Ecosystem consequences of bird declines. *Proc Natl Acad Sci US A*, 101(52):18042–18047, 2004.

[39] A. Selin, J. Turunen, and J. T Tanttu. Wavelets in recognition of bird sounds. *EURASIP Journal on Applied Signal Processing*, 2007(1):141–141, 2007.

[40] S. A Selouani, M. Kardouchi, E. Hervet, and D. Roy. Automatic birdsong recognition based on autoregressive time-delay neural networks. In *Computational Intelligence Methods and Applications, 2005 ICSC Congress on*, pages 6–pp.

[41] P. Somervuo, A. Harma, and S. Fagerlund. Parametric representations of bird sounds for automatic species recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(6):2252–2263, 2006.

[42] M. Thangavelu and R. Raich. Multiclass linear dimension reduction via a generalized chernoff bound. In *Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on*, pages 350–355. IEEE, 2008.

[43] Vlad M. Trifa, Alexander N. G. Kirschel, Charles E. Taylor, and Edgar E. Vallejo. Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *The Journal of the Acoustical Society of America*, 123(4):2424, 2008.

[44] E. Vilches, I. A Escobar, E. E Vallejo, and C. E Taylor. Data mining applied to acoustic bird species recognition. *Pattern Recognition*, 3:400–403, 2006.

[45] Y. Wang. Distributed gibbs sampling of latent dirichlet allocation: The gritty details.

APPENDICES

A First appendix - Detailed results for section 8

Table 4: Estimation of θ

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc7	Doc 8	Doc 9
θ_1	0.0095	0.0847	0.0913	0.0489	0.2269	0.1187	0.4287	0.0330	0.0106
θ_2	0.0195	0.0000	0.0593	0.1061	0.0634	0.0000	0.0000	0.0211	0.0051
θ_3	0.1609	0.0000	0.0494	0.0147	0.0075	0.0877	0.0005	0.0004	0.0188
θ_4	0.0180	0.2027	0.1410	0.0592	0.0003	0.0000	0.0000	0.0102	0.1063
θ_5	0.0000	0.4496	0.0000	0.0427	0.1267	0.0000	0.0333	0.0448	0.3445
θ_6	0.1046	0.0000	0.0000	0.0112	0.0000	0.2328	0.0911	0.3323	0.0000
θ_7	0.0000	0.0000	0.1999	0.0932	0.0298	0.0562	0.0019	0.0000	0.0308
θ_8	0.0001	0.0000	0.0006	0.1237	0.0000	0.0059	0.0167	0.0000	0.0000
θ_9	0.3391	0.0078	0.0666	0.0124	0.0391	0.0357	0.0103	0.1574	0.0000
θ_{10}	0.0000	0.0075	0.0748	0.2750	0.1540	0.0182	0.2327	0.1241	0.0000
θ_{11}	0.0905	0.0570	0.0000	0.0000	0.0105	0.0598	0.0000	0.0733	0.0005
θ_{12}	0.0000	0.0102	0.0202	0.0716	0.0802	0.0762	0.0001	0.0000	0.3156
θ_{13}	0.2121	0.0000	0.0196	0.0699	0.0000	0.0000	0.1420	0.0469	0.0826
θ_{14}	0.0458	0.0000	0.0000	0.0275	0.0000	0.0000	0.0000	0.1564	0.0526
θ_{15}	0.0000	0.0000	0.0006	0.0420	0.2582	0.0017	0.0000	0.0000	0.0327
θ_{16}	0.0000	0.1806	0.2768	0.0018	0.0035	0.3069	0.0427	0.0000	0.0000

Table 5: Γ matrix

	Z=1	Z=2	Z=3	Z=4
S_1	1	0	0	0
S_2	1	0	0	0
S_3	1	0	0	0
S_4	1	0	0	0
S_5	0	1	0	0
S_6	0	1	0	0
S_7	0	1	0	0
S_8	0	1	0	0
S_9	0	0	1	0
S_{10}	0	0	1	0
S_{11}	0	0	1	0
S_{12}	0	0	1	0
S_{13}	0	0	0	1
S_{14}	0	0	0	1
S_{15}	0	0	0	1
S_{16}	0	0	0	1

Table 6: Estimation of ϕ

v	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
1	0.0704	0.1712	0.1779	0.2144	0.0692	0.0643	0.0244	0.046
2	0	0	0	0	0	0	0	0.0006
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0.0002	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0.0008	0.0007	0	0.0041	0.0146	0.0065	0.0153	0.012
8	0.0036	0.0003	0.0003	0	0.0255	0.0031	0.0228	0.0385
9	0.0002	0.0003	0	0.0014	0	0	0	0
10	0	0	0	0	0.0003	0	0	0
11	0	0	0	0.0001	0	0	0	0
12	0	0	0	0	0.0774	0.0631	0.026	0.0089
13	0	0	0.0001	0	0.0002	0	0	0.0002
14	0	0	0	0	0.0043	0.0108	0	0
15	0	0	0	0	0	0.0001	0	0
16	0	0	0	0.0002	0	0.0002	0	0
17	0	0	0	0	0.0034	0.0024	0.0018	0
18	0.005	0.0123	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0.0006	0	0.0002	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0.0012	0.0001	0	0	0
23	0.0821	0.0978	0.033	0.0609	0.0646	0.0177	0.0398	0.0566
24	0	0	0	0	0.0201	0.0063	0.0087	0.0465
25	0	0	0	0	0	0	0	0
26	0.0338	0	0.0592	0.0296	0.0003	0.0002	0.0002	0
27	0.0012	0.0002	0.0071	0	0	0	0	0.0002
28	0.0014	0.0004	0.0026	0	0	0	0	0
29	0	0	0	0	0	0	0.001	0.0002
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0.0074	0.0038	0.0004	0.0049
32	0.0024	0	0	0	0	0	0	0
33	0	0.0013	0	0	0.0001	0	0	0
34	0	0	0	0	0.075	0.0168	0.08	0.119
35	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0.0032	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0.0048	0	0	0	0.0008	0.0002
40	0	0	0	0.0003	0.0009	0	0	0
41	0	0	0	0	0.0017	0.0046	0	0
42	0	0	0	0	0	0	0	0.0001
43	0.0073	0.0179	0.0057	0.0042	0	0	0	0
44	0	0	0	0	0	0	0	0
45	0	0	0	0	0.0051	0.0092	0	0.0056
46	0	0	0.0046	0	0	0	0	0
47	0.0634	0.0359	0.0659	0.0242	0.0131	0.0189	0.0065	0.0004
48	0	0	0	0	0.0003	0.0074	0	0.0046
49	0	0	0	0	0.0111	0.004	0	0

Table 7: Estimation of ϕ - Continuing

$v=1$	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
1	0	0.0076	0.0025	0.001	0.0003	0	0	0.0021
2	0.0231	0.0191	0	0.0234	0.06	0.0652	0.0183	0.0187
3	0	0	0	0.0084	0.0004	0.0001	0	0
4	0	0	0	0	0.0016	0.0089	0	0.0102
5	0.1303	0.1629	0.2025	0.0507	0.0539	0.097	0.045	0.0293
6	0	0.0002	0.004	0.0027	0	0	0	0
7	0	0.0003	0	0	0	0	0.0004	0
8	0	0	0	0	0.0001	0	0	0
9	0	0	0	0	0	0	0	0
10	0.0133	0	0.0107	0.0006	0	0	0.0012	0
11	0	0.0003	0	0	0	0	0.0001	0.0022
12	0.1476	0.3746	0.2019	0.2235	0.0014	0.0006	0.0016	0.005
13	0	0	0	0	0	0	0	0.0002
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0.0001	0
16	0.1934	0.0906	0.1701	0.1303	0	0	0	0
17	0	0.0017	0	0.0011	0	0	0	0
18	0.0124	0.0193	0.0105	0.011	0	0	0	0
19	0	0	0	0	0.0429	0.0225	0.0685	0.0725
20	0.0017	0	0	0.0026	0	0	0	0
21	0	0.0021	0.0002	0.0061	0	0	0	0
22	0	0.0004	0	0	0.1014	0.0475	0.1129	0.1365
23	0	0	0.0013	0	0.0106	0.0051	0.0175	0.0282
24	0	0	0	0	0	0	0	0
25	0.0018	0	0	0	0	0	0	0
26	0.0117	0.0366	0	0.0297	0	0.0058	0	0.0003
27	0.0853	0.0303	0.0899	0.0573	0	0	0	0.0005
28	0	0	0	0.0001	0	0.0059	0.0036	0
29	0	0	0	0	0	0	0	0
30	0	0.0002	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0
35	0.0021	0	0	0	0.0054	0.0177	0.0064	0.0064
36	0	0	0	0	0.0239	0.0235	0.0412	0.0023
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0.2547	0.1018	0.0869	0.1987
41	0	0	0	0	0.0001	0	0	0
42	0.0074	0.0002	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0.0011	0	0
45	0	0	0.0004	0.0005	0	0.0008	0	0
46	0	0	0	0	0	0	0	0
47	0	0.0002	0	0	0	0	0	0
48	0	0	0	0	0.0004	0	0	0.0012
49	0.0101	0	0.0378	0.0074	0	0	0	0

Table 8: Estimation of ϕ - Continuing

v	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
50	0.0235	0.0175	0	0.0155	0	0	0	0
51	0.0003	0.0007	0.0002	0	0	0	0	0
52	0.0131	0.0048	0.0295	0.0027	0	0	0	0
53	0.0695	0.0989	0.1071	0.1533	0.0001	0	0	0.0004
54	0.0404	0.0373	0	0.0241	0	0	0.0071	0
55	0	0	0	0.0002	0.0287	0.0259	0.0268	0.0486
56	0.0069	0	0.0294	0.003	0.1169	0.1138	0.1487	0.2069
57	0.0464	0.0569	0.1266	0.1122	0.0122	0.01	0	0.0083
58	0.0002	0	0.0001	0	0	0.0002	0.0002	0
59	0	0	0.0024	0.0173	0.025	0.0197	0.0251	0.0292
60	0.0002	0	0.0001	0	0.0006	0	0.0003	0
61	0	0	0.0001	0	0.0002	0	0	0
62	0	0	0	0	0	0	0	0
63	0	0	0.0002	0.0008	0	0	0	0
64	0	0	0	0	0	0	0	0.0015
65	0	0	0	0.0003	0	0	0	0
66	0	0	0	0	0.0001	0.0002	0.0001	0
67	0	0	0	0	0.0107	0	0.0004	0.0003
68	0	0	0	0.0035	0	0	0	0
69	0	0	0.0003	0	0	0	0	0
70	0.0039	0	0.0002	0.0002	0	0.0001	0	0
71	0.0003	0	0	0	0.0332	0.01	0.0387	0.0483
72	0	0	0	0	0.1295	0.1298	0.0528	0.0399
73	0.0134	0.0015	0.0007	0	0	0	0	0
74	0	0	0	0	0	0	0	0.0001
75	0.0327	0.0041	0.0142	0.0406	0	0.0004	0.005	0.0002
76	0	0	0	0	0.0313	0.0059	0.0466	0.0379
77	0.1512	0.2168	0.0149	0.0826	0	0	0	0
78	0.0275	0.0324	0.0006	0.009	0	0	0	0
79	0	0.0025	0	0	0.0264	0.0286	0.0214	0.0532
80	0.0073	0	0.0011	0	0	0.0002	0	0
81	0	0.0004	0.0009	0.0017	0.0001	0	0	0
82	0.0249	0.0398	0.0081	0.012	0.0095	0.0304	0.0367	0.0416
83	0	0	0	0	0	0	0	0
84	0	0.0017	0.0001	0.0031	0.0516	0.0128	0.0577	0.0358
85	0	0	0.0001	0	0	0	0	0
86	0	0	0	0	0	0.0007	0	0
87	0	0	0	0	0	0	0	0
88	0.0055	0.009	0.0334	0.0298	0	0	0	0
89	0	0	0	0	0.054	0.1637	0.1595	0.0391
90	0	0	0	0	0	0	0.0001	0
91	0.0029	0.0269	0	0.0275	0	0	0	0
92	0	0	0	0	0	0	0	0
93	0.0454	0.0393	0.0556	0.0462	0	0.0008	0	0.001
94	0.002	0	0	0.0059	0	0	0	0
95	0.0075	0	0	0	0	0	0	0
96	0	0	0	0	0	0.0001	0.0006	0.0045
97	0	0.0003	0.0091	0.0043	0	0	0	0
98	0	0	0.0001	0	0.0442	0.0506	0.0379	0.0135
99	0.0079	0	0	0	0.0222	0.0964	0.0598	0.025
100	0.1955	0.0704	0.2021	0.063	0.008	0.0564	0.0461	0.0194

Table 9: Estimation of ϕ - Continuing

v	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
50	0	0	0	0.0001	0.0002	0	0	0.0000
51	0.0002	0.004	0.0043	0.0083	0	0	0	0.0000
52	0	0	0	0	0	0	0.0002	0.0002
53	0.0527	0.0498	0.0353	0.056	0.0024	0.0026	0.0002	0.0023
54	0	0	0	0	0.0039	0	0	0.0000
55	0	0	0	0	0	0.0098	0.0111	0.0202
56	0	0	0	0	0.0039	0	0	0.0000
57	0	0	0	0	0.0456	0.0186	0.0272	0.0471
58	0	0.0115	0.0127	0	0	0	0	0.0000
59	0.0001	0	0	0	0	0	0	0.0000
60	0.0095	0.0102	0	0.0233	0	0	0	0.0000
61	0	0	0	0	0.0809	0.0386	0.013	0.0396
62	0	0.0001	0	0.0018	0.0174	0.0316	0.0728	0.0070
63	0.0001	0	0	0	0	0	0.0043	0.0003
64	0.0006	0.0006	0	0	0	0	0	0.0000
65	0	0	0	0	0.0663	0.0577	0.0794	0.1537
66	0	0	0	0	0.001	0	0	0.0001
67	0	0	0	0	0	0	0	0.0000
68	0	0.0063	0.0066	0.0076	0	0	0.0002	0.0000
69	0	0	0	0	0	0	0	0.0000
70	0	0	0	0	0	0	0	0.0000
71	0	0	0	0	0	0	0	0.0000
72	0.0032	0.0018	0.005	0	0.0204	0.0471	0.0463	0.0296
73	0	0	0	0	0	0	0	0.0000
74	0.1717	0.1072	0.0682	0.2444	0.0664	0.1068	0.0495	0.0486
75	0	0	0	0.0003	0.0123	0.0032	0	0.0005
76	0.0291	0.0062	0.0088	0.0005	0.0066	0.0361	0.0003	0.0007
77	0	0	0	0	0.0005	0	0	0.0000
78	0	0	0	0	0	0	0	0.0000
79	0.0292	0.002	0.0382	0.033	0.0123	0.0386	0.0184	0.0106
80	0.0003	0.0008	0	0.0003	0.0224	0.0259	0.024	0.0329
81	0	0.0008	0	0.0008	0	0	0	0.0045
82	0	0	0	0	0	0.0038	0	0.0018
83	0.0263	0.03	0.0368	0.0116	0	0.001	0.0005	0.0003
84	0	0	0	0	0	0	0	0.0000
85	0.0039	0	0	0.0039	0	0	0	0.0000
86	0.0004	0	0	0	0.0007	0	0	0.0000
87	0	0	0	0	0	0	0	0.0001
88	0	0	0	0	0	0	0	0.0000
89	0	0	0	0	0.0737	0.1592	0.2474	0.0765
90	0	0	0	0	0.0009	0.005	0	0.0023
91	0	0	0	0	0	0	0	0.0000
92	0	0.0002	0.0002	0	0	0	0	0.0000
93	0.0003	0	0	0	0	0.0002	0	0.0015
94	0	0.0005	0	0	0	0	0	0.0000
95	0	0	0	0	0	0	0.0002	0.0000
96	0	0	0	0	0	0	0	0.0000
97	0	0.0001	0	0	0.0043	0	0.0005	0.0000
98	0	0	0	0	0	0.005	0	0.0036
99	0	0	0	0	0	0.0013	0	0.0005
100	0.0319	0.0205	0.0165	0.0504	0	0.0026	0.0002	0.0000

Figure 15: Sampled ϕ at iteration 500

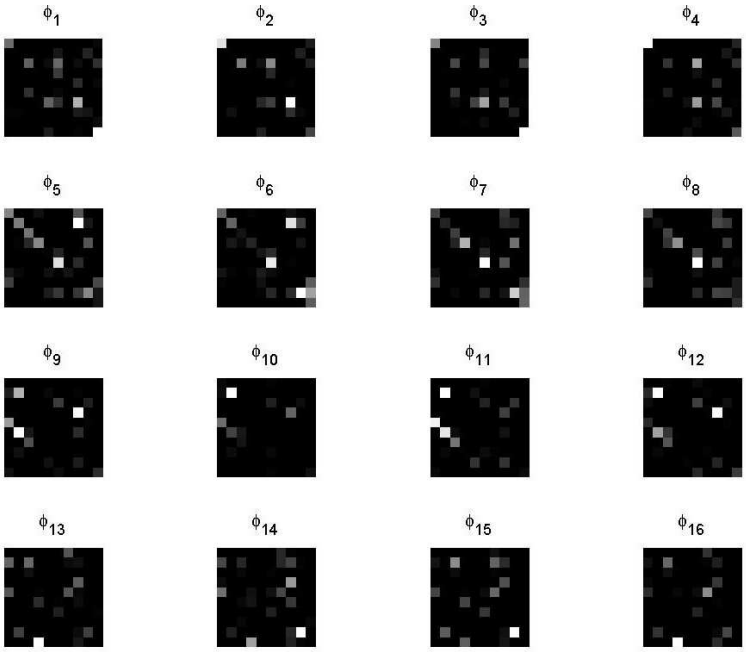


Figure 16: Sampled ϕ at iteration 600

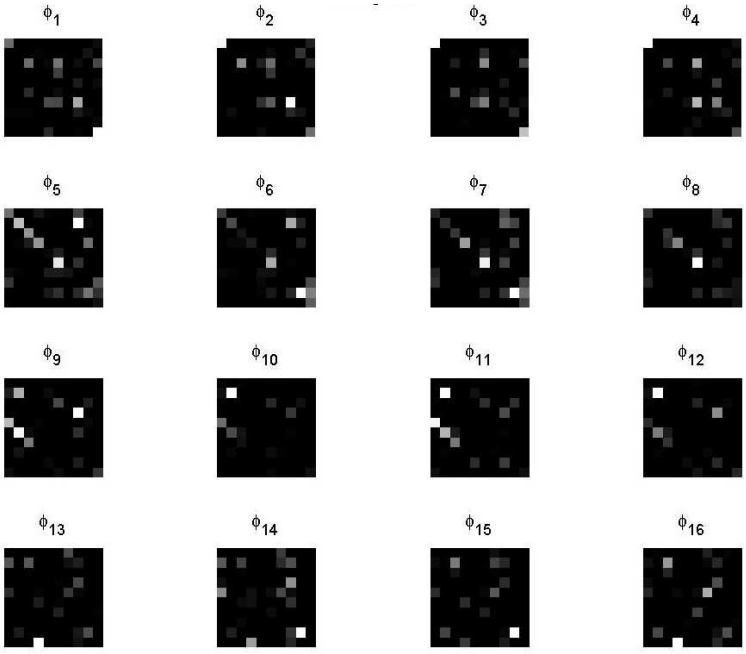


Figure 17: Sampled ϕ at iteration 700

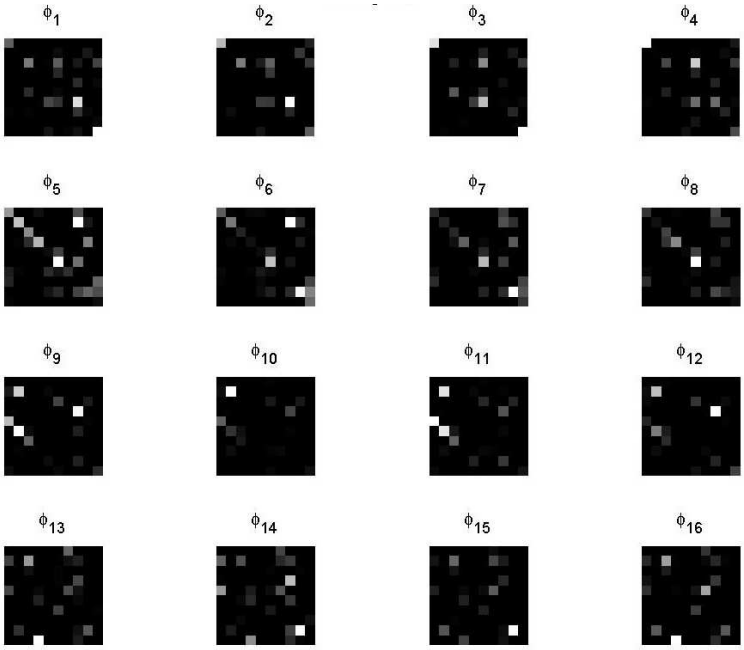


Figure 18: Sampled ϕ at iteration 800

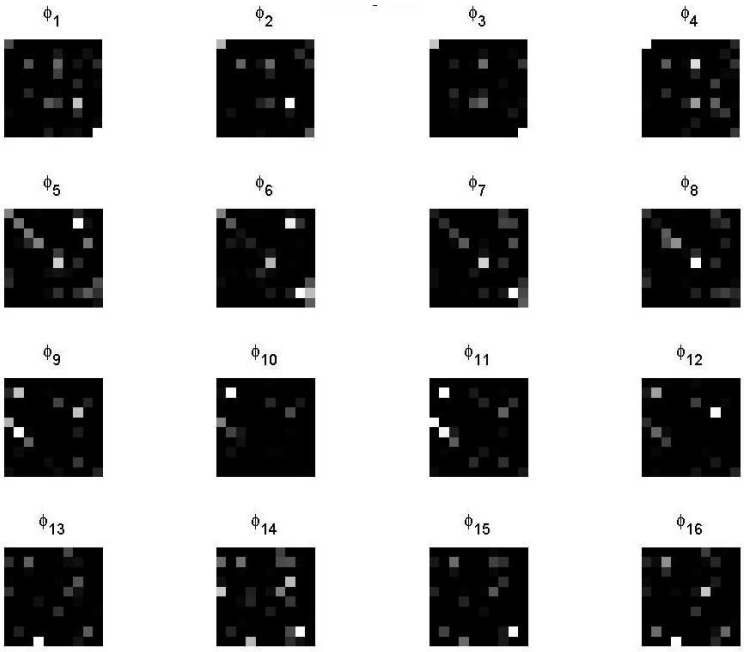


Figure 19: Sampled ϕ at iteration 900

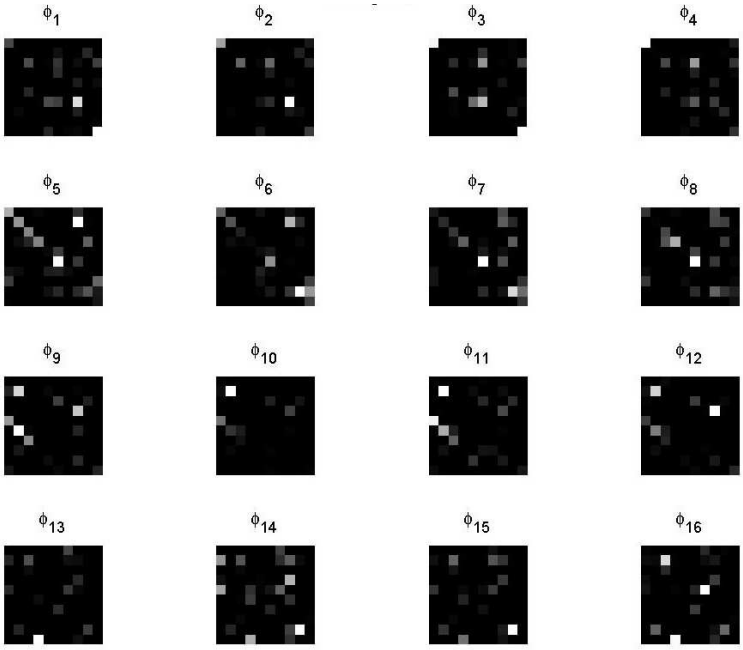
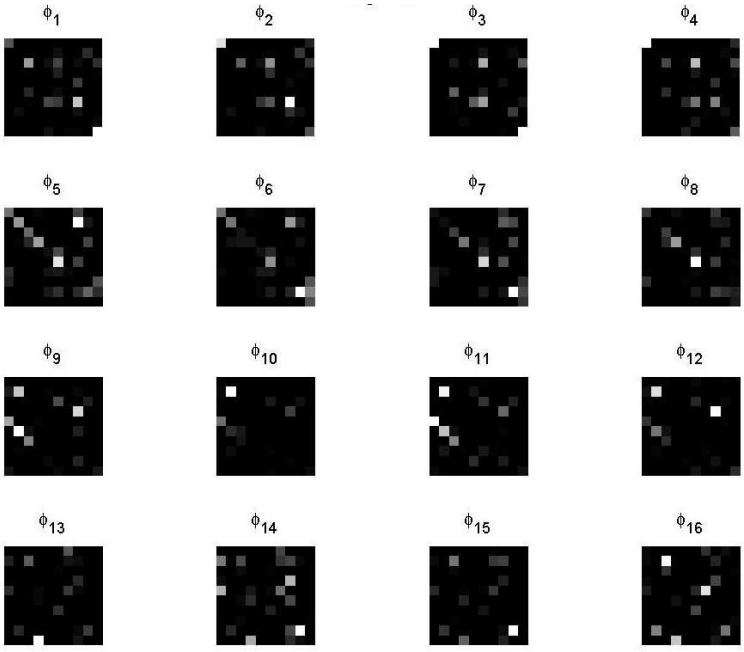


Figure 20: Sampled ϕ at iteration 1000



B Second appendix - Apriori Algorithm Results

```
Input configuration: 13 items, 548 transactions, minsup = 5.0%
Apriori algorithm has started.
Frequent 1-itemsets
[1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13]
Frequent 2-itemsets
[1 2, 1 3, 1 4, 1 6, 2 3, 2 13, 3 13, 4 6, 9 10, 9 11, 10 11]
Frequent 3-itemsets
[1 2 3, 1 4 6, 2 3 13, 9 10 11]
Execution time is: 0.019 seconds.
```

A	B	Confidence/Precision
'1'	'2'	0.345
'2'	'1'	0.624
'1'	'3'	0.579
'3'	'1'	0.691
'1'	'4'	0.269
'4'	'1'	0.646
'1'	'6'	0.376
'6'	'1'	0.822
'2'	'3'	0.881
'3'	'2'	0.582
'2'	'13'	0.376
'13'	'2'	0.891
'3'	'13'	0.261
'13'	'3'	0.935
'4'	'6'	0.573
'6'	'4'	0.522
'9'	'10'	0.708
'10'	'9'	1.000
'9'	'11'	0.461
'11'	'9'	0.519
'10'	'11'	0.460
'11'	'10'	0.367
'1 2'	'3'	0.882
'1 3'	'2'	0.526
'1'	'2 3'	0.305
'2 3'	'1'	0.625
'2'	'1 3'	0.550
'3'	'1 2'	0.364
'1 4'	'6'	0.642
'1 6'	'4'	0.459

A	B	Confidence/Precision
'1'	'4 6'	0.173
'4 6'	'1'	0.723
'4'	'1 6'	0.415
'6'	'1 4'	0.378
'2 3'	'13'	0.396
'2 13'	'3'	0.927
'2'	'3 13'	0.349
'3 13'	'2'	0.884
'3'	'2 13'	0.230
'13'	'2 3'	0.826

'9 10'	'11'|	0.460|
'9 11'	'10'|	0.707|
'9'	'10 11'|	0.326|
'10 11'	'9'|	1.000|
'10'	'9 11'|	0.460|
'11'	'9 10'|	0.367|