AN ABSTRACT OF THE THESIS OF

Balaji Lakshminarayanan for the degree of Master of Science in
Electrical & Computer Engineering presented on December 07, 2010.
Title: Probabilistic Models for Classification of Bioacoustic Data.

Abstract approved: _____

Raviv Raich

Probabilistic models have been successfully applied for a wide variety of problems, such as but not limited to information retrieval, computer vision, bio-informatics and speech processing. Probabilistic models allow us to encode our assumptions about the data in an elegant fashion and enable us to perform machine learning tasks such as classification and clustering in a principled manner. Probabilistic models for bio-acoustic data help in identifying interesting patterns in the data (for instance, the species-specific vocabulary), as well as species identification (classification) in recordings where the label is not available.

The focus of this thesis is to develop efficient inference techniques for existing models, as well as develop probabilistic models tailored to bioacoustic data. First, we develop inference algorithms for the supervised latent Dirichlet allocation (LDA) model. We present collapsed variational Bayes, collapsed Gibbs sampling and maximum-a-posteriori (MAP) inference for parameter estimation and classification in supervised LDA. We provide an empirical evaluation of the trade-off between computational complexity and classification performance of the inference methods for supervised LDA, on audio classification (species identification in this context)

as well as image classification and document classification tasks. Next, we present novel probabilistic models for bird sound recordings, that can capture temporal structure at different hierarchical levels, and model additional information such as the duration and frequency of vocalizations. We present a non-parametric density estimation technique for parameter estimation and show that the MAP classifier for our models can be interpreted as a weighted nearest neighbor classifier. We provide an experimental comparison between the proposed models and a support vector machine based approach, using bird sound recordings from the Cornell Macaulay library.

Probabilistic Models for Classification of Bioacoustic Data

by

Balaji Lakshminarayanan

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 07, 2010
Commencement June 2011

Master of Science thesis of Balaji Lakshminarayanan presented on December 07, 2010

APPROVED:

Major Professor, representing Electrical & Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Balaji Lakshminarayanan, Author

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Dr. Raviv Raich for being such a wonderful mentor. Dr. Raich gave me the opportunity to work on many interesting problems, guided me in the right direction and always made himself available for discussions in spite of his busy schedule. His contagious enthusiasm and commitment towards research are some of the qualities that I would like to emulate. I thank Dr. Robert Smythe and Dr. Xiaoli Fern for agreeing to be on my committee. I was introduced to many concepts in statistics and machine learning through their courses. My discussions with them were very useful for my research. I thank Dr. Yevgeniy Kovchegov for his valuable time.

I thank all the members of the Bioacoustics group for stimulating discussions and helpful feedback. Thanks to Forrest Briggs for allowing me to use his data for my experiments. I would also like to thank Dr. Cedric Archambeau and Dr. Guillaume Bouchard of Xerox Research Center Europe. Though not directly related to this thesis, my interactions with them greatly improved my understanding of graphical models. I thank all my group members, especially Madan, Deepthi, Behrouz and Ali for numerous helpful discussions and all the fun.

I am grateful to my family members for their love and support and more importantly, their patience. I am lucky to have some awesome friends (you know who you are ☺), thank you folks, for your timely support, excellent sense of humor and all the crazy discussions. Special thanks to Krishna for being such an amazing friend.

CONTRIBUTION OF AUTHORS

The independent block model in Chapter 3. was developed in collaboration with Dr. Xiaoli Fern and Forrest Briggs, unifying the models developed previously in [1, 2]. Forrest Briggs provided the Mel frequency cepstral coefficients data, the codebook representation of all the frame level features (mean frequency + bandwidth, Mel frequency cepstral coefficients, normalized power spectral density) for the bird sound recordings, and the results for the Multinomial Interval IID model.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and Background

1.1.1 Landscape bioacoustics

Landscape bioacoustics refers to the study of large-scale ecological processes through computational analysis of biological sounds. In this work, we focus specifically on bird sounds. Landscape bioacoustics can address important ecological questions such as

- How do bird populations vary as a function of vegetation, climate, time?
- Do birds vocalize differently (eg., vary the vocalization frequency) based on presence/absence of other species of birds?
- How do the vocalization patterns from a single species vary with respect to parameters such as age, geographical location?

Existing bird species distribution data are collected by human effort, which can be expensive, labor intensive, and subject to human bias. Our goal is to develop probabilistic models and efficient inference techniques for automatically identifying the bird species present in an audio recording. Such algorithms will serve as part of a system to automatically collect bird species presence/absence data, which will provide valuable ecological information for species distribution modeling and conservation planning [2]. Automatic bird species identification will enable us to track bird populations across varying landscapes, with an unprecedented time resolution, in a cost effective manner. Additionally, probabilistic models help us understand the vocalization patterns of individual bird species and learn the 'language of birds'.

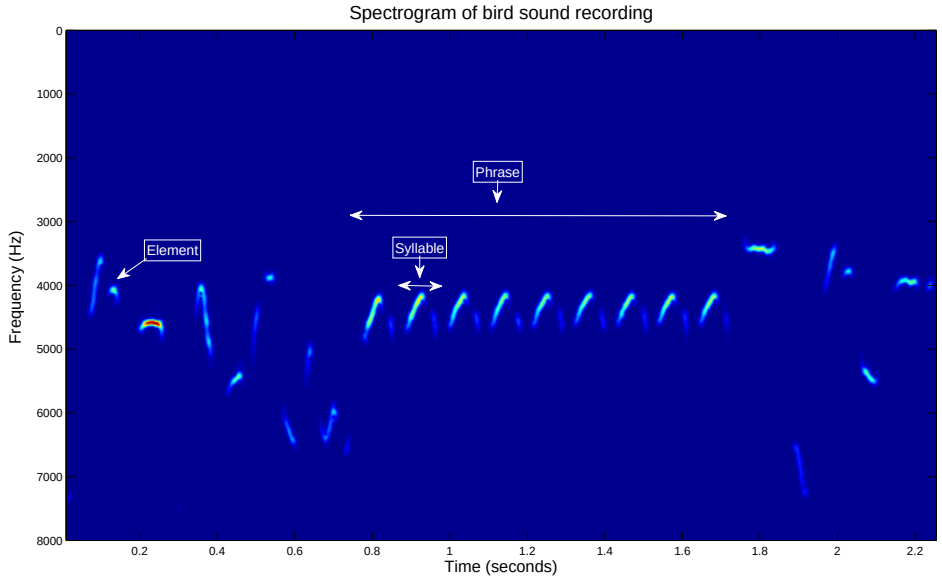
The members of the Bioacoustics research group at Oregon State university have deployed song meters (recording devices) at various locations in the H.J. Andrews (HJA) experimental forest during Summer 2009, 2010 and collected more than a tera byte of audio data. Most of the recordings are not labeled; the recordings contain simultaneous vocalizations of multiple bird species. The recordings are plagued by various sources of noise such as stream noise, rain fall, other animal species/insects in the forests. It is expensive to obtain detailed human annotation (i.e., labels for each of the constituent elements) for such recordings. Automated analysis of such large scale, noisy data presents significant challenges to current machine learning algorithms. Some of the interesting questions are:

- What is the appropriate feature representation for a bird sound recording?
- How do we learn the vocabulary and song structure (if any) of each bird species? For other types of audio sounds (eg., human speech), knowledge of the vocabulary and the grammatical structure of the language may be employed to improve the performance of the speech recognition system. However, such information is not available for bird sounds.
- How do we learn classification rules from limited supervision? (eg., just information about presence/absence of species in audio recordings, instead of labeling each of the constituent elements in the recording)
- How can we identify individuals based on limited training data from that species? Developing models to answer this question will help in obtaining abundance counts, rather than just presence/absence of bird species.

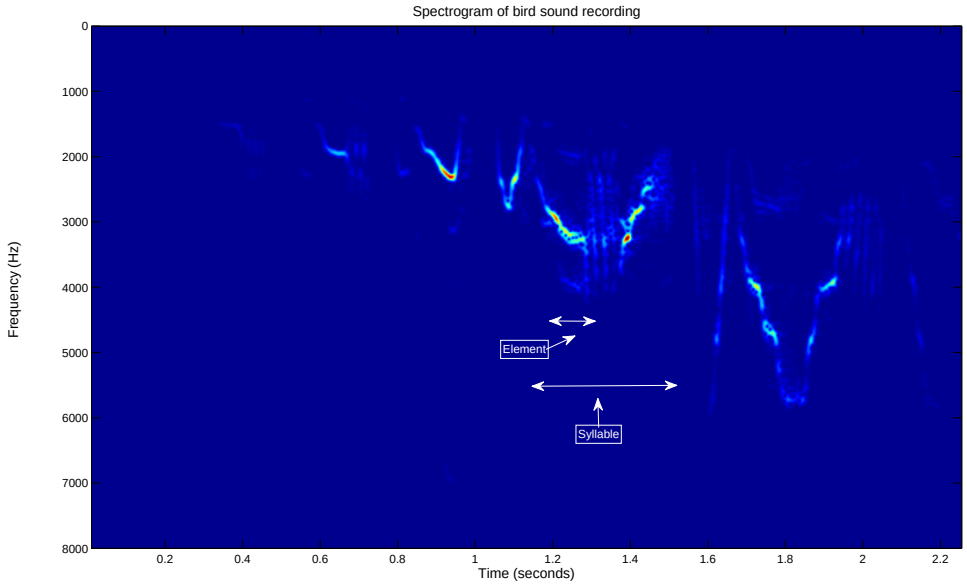
This work is a first step towards solving the ambitious goals stated above. Here, we assume that each recording is labeled with a single species, which usually corresponds to the most prominent bird species within the recording. We develop probabilistic models that can identify the species present in a new test recording, as well as learn the patterns of the vocalizations of each bird species present in the training data. We develop efficient inference algorithms for our models and evaluate the classification accuracy of the different approaches using recordings from the Cornell Macaulay library.

1.1.2 Nature of bird vocalization

Human speech recognition systems are generally based on models that characterize the vocabulary and grammar of a particular language. The notion of vocabulary and grammar is usually ambiguous for sound signals other than human speech. However, bird vocalization is a good example of a class of natural sounds where we can expect to find an underlying vocabulary and inherent grammatical structure [3]. Bird recordings usually contain a structured pattern of brief sounds from a species-specific vocabulary. Fig. 1.1(a) and 1.1(b) display the spectrogram of two recordings containing the species Winter Wren and Swainson's Thrush respectively. Bird sounds can be hierarchically sub divided into the following levels: phrases, syllables and elements [4]. Elements consist of multiple frames, where frame corresponds to a very short interval of time (roughly milli seconds). Syllables are constructed of one or more elements, but they can be seen as suitable unit for recognition of bird species because they can be more reliably detected from continuous recordings than elements [5]. Phrases capture the patterns of syllables and usually include more regional and individual variability than syllables [5].



(a)



(b)

FIGURE 1.1: Spectrogram of a recording belonging to (a) Winter Wren species (b) Swainson's Thrush species.

1.1.3 Bird species identification system

The basic block diagram of a typical bird species identification system is shown in Fig. 1.2. In the training stage, we observe the audio recordings along with the corresponding label for each recording. Most systems typically compute the spectrogram of the audio recording, process the spectrogram and then obtain a feature vector representation for each recording. The <`feature vector, label`> pairs for the training documents are used as input to a machine learning algorithm that can learn a classification rule. In the test stage, we obtain the feature vector representation for the audio recording and predict the label of the test recording. In this work, we focus on developing probabilistic models that learn a model based on the training data and predict the species present in the test data based on the model.

1.2. Feature vector representation of the audio signal

Audio classification systems typically begin by extracting acoustic features from audio signals. Such features often pertain to individual frames (i.e., very short duration of signal). Each sound recording contains multiple syllables and each syllable in turn, contain multiple frames. To apply many standard algorithms for classification, it is necessary to represent a sound recording using a set of fixed-length vectors. In this section, we present a general introduction to the methods. Exact parameter settings and other details about our implementation are available in [6].

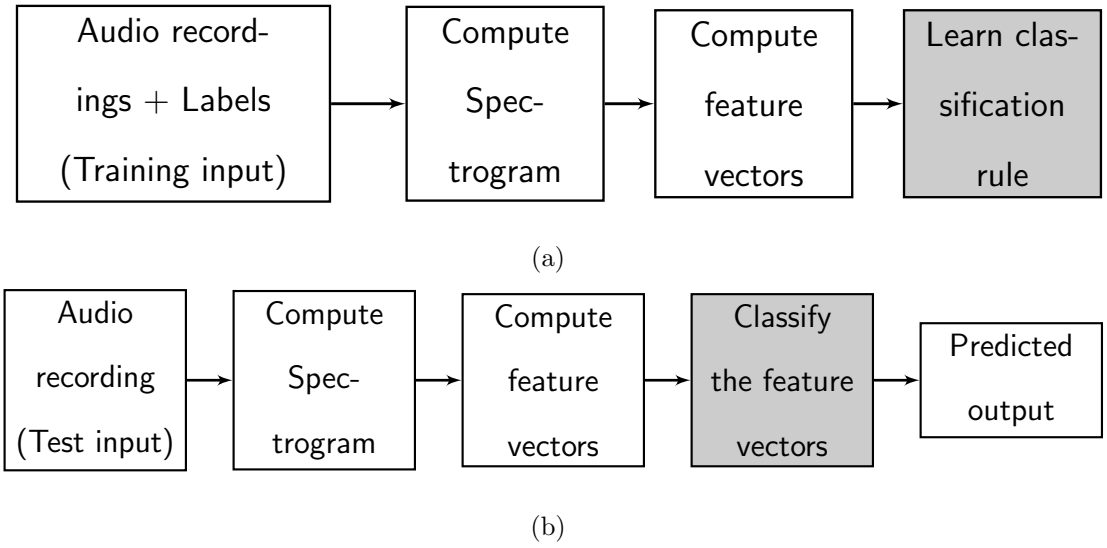


FIGURE 1.2: Typical Bird species identification system (a) Training stage: The labels (i.e., the list of species present in the recording) are observed for training data (b) Test stage: The labels are predicted based on the classification rule learnt from the training data. In this work, we develop probabilistic models to learn from training data and classify test data (shaded boxes).

1.2.1 Spectrogram

Throughout this work, we will use the spectrogram of an audio signal for various purposes. In this section, we will briefly review the computation and interpretation of spectrograms. Fig. 1.3 shows the time domain representation of a chirp signal. The basic idea of a spectrogram is to divide a time domain signal into overlapping frames (windows in time domain) and compute the frequency spectrum of each frame. The magnitude of the frequency spectrum corresponds to the intensity of the image (blue corresponds to lower intensity, red corresponds to higher intensity). Fig. 1.4 provides the time domain signals and the corresponding spectrograms of chirps. We can observe the correspondence between the rate of increase (decrease) in the time domain signal and the variation of the magnitude spectrogram. To compute the spectrogram, we need to define the following parameters

- Frame (Time window) size
- Discrete Fourier Transform size (number of frequency bins)
- Amount of overlap between successive frames

Essentially, the spectrogram provides a visual time-frequency representation of the audio signal.

1.2.2 Segmentation of the audio recording

A typical pre-processing step for audio classification is segmentation, which is the process of identifying the interesting segments present in a recording. Typically, the interesting segments correspond to the syllables produced by the birds. Segmentation is inherently an ill-posed problem and we typically obtain only a list

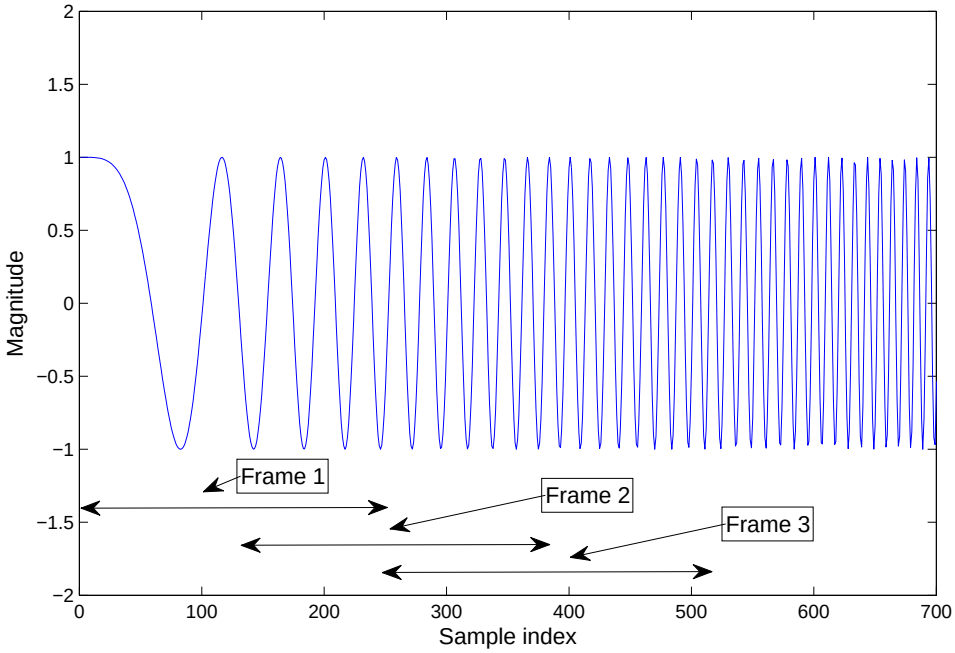
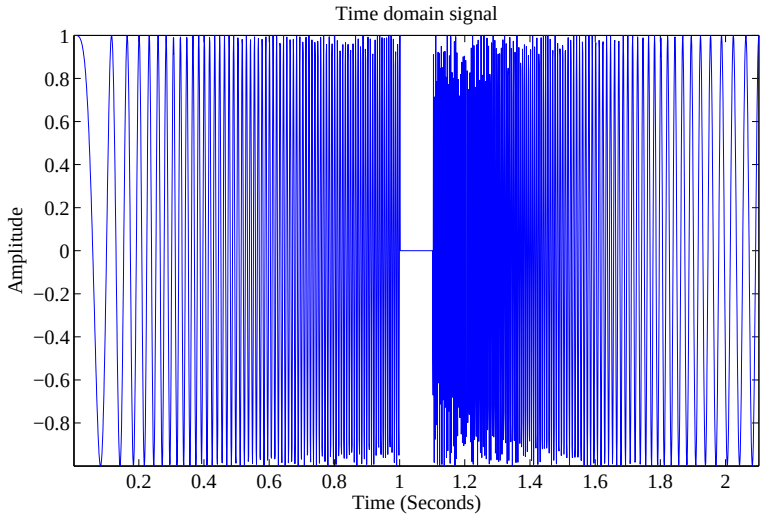
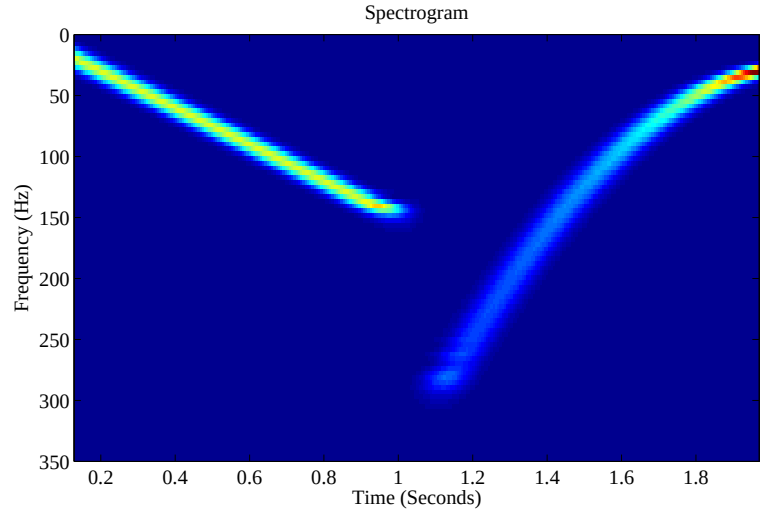


FIGURE 1.3: Spectrogram computation: The time domain signal is subdivided into frames, with possible overlap between successive frames and the frequency spectrum of the frames is computed using the Discrete Fourier Transform.



(a)



(b)

FIGURE 1.4: Interpreting a spectrogram: (a) The time domain signal (b) The corresponding spectrogram. The spectrogram provides a time-frequency visualization of the audio signal i.e., the change in frequency in the original signal can be captured by tracking the magnitude of the spectrogram as a function of time and frequency.

of potential syllables. Once we identify the frames corresponding to the list of potential syllables, we may aggregate the frame level features at the syllable level or we may aggregate the frame level features for the entire recording [3, 7, 8].

Segmentation algorithms can be broadly classified into

- Time domain segmentation (1-D): Here the segmentation is done only in the time domain, i.e., the segmentation algorithm returns a set of vertical markers in the spectrogram.

- Time-frequency segmentation (2-D): Here the segmentation is done in both time and frequency domain, i.e., the segmentation algorithm returns a set of regions enclosing the syllables in the spectrogram. A simple example would be boxes around the syllables, but the regions can assume free-form shapes in general.

2-D segmentation algorithms are helpful when multiple sources of sound (corresponding to different frequencies) are present at the same time instant. These segments can be used as the building blocks for the classifier. For instance, the segments can be used as the instances in an multiple instance multiple label (MIML) algorithm. Next, we illustrate some approaches for segmentation.

Manual segmentation

Here, we manually create the segmentation boundaries. The boundaries may be free form shapes, boxes around the syllables, or vertical markers in the time domain (1-D). Fig. 1.5 illustrates a 2-D manual segmentation algorithm where the segmentation boundaries are marked by boxes. Manual segmentation can be expensive and tedious.

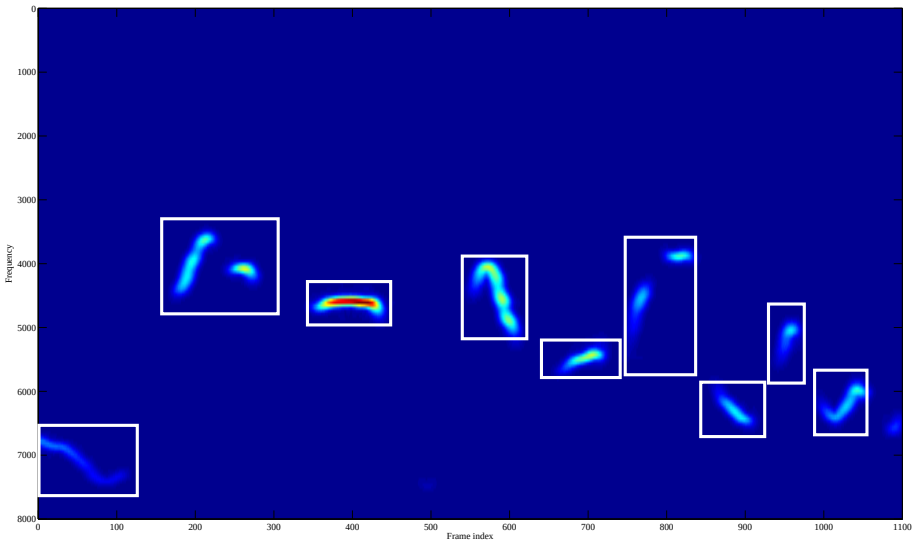


FIGURE 1.5: Example of 2-D Manual segmentation: Boxes correspond to the segmentation boundaries. In general, the segmentation boundaries may be free form shapes.

Energy based segmentation

First, we describe a 1-D energy segmentation algorithm. We first compute the spectrogram of the audio signal and then compute the energy of each frame. We determine a threshold for energy of frames corresponding to syllables. Next, we determine the points where the energy of the signal intersects the threshold. If the derivatives at these points have positive (negative) sign, these are marked as the beginning (end) of the syllable. Fig. 1.6 illustrates this approach. Instead of fixing a single threshold, we can initially start with a high threshold, identify the elements corresponding to noise and signal, re-compute the threshold, and iteratively identify the final list of syllables. An alternative energy based segmentation procedure is described in Fig. 1.7. First, we determine sharp local minima in the energy. These endpoints correspond to transitions between high vocalization and low vocalization, and are potential syllable markers. The regions between these endpoints are treated as potential syllables and the energy of each potential syllable is computed. Based on a threshold, we decide the final list of syllables. Sometimes, it may be beneficial to merge adjacent elements to obtain longer syllables. A similar approach may be used for 2-D energy based segmentation.

KL-divergence based segmentation

We first compute the spectrogram of the audio signal and then compute the power spectral density (PSD) of each frame and normalize it to obtain the normalized PSD. Note that the normalized PSD is a valid probability density function. Next, we compute the Kullback Leibler (KL) divergence between the normalized power spectral density (PSD) of each frame and the uniform distribution. Syllables contain spec-

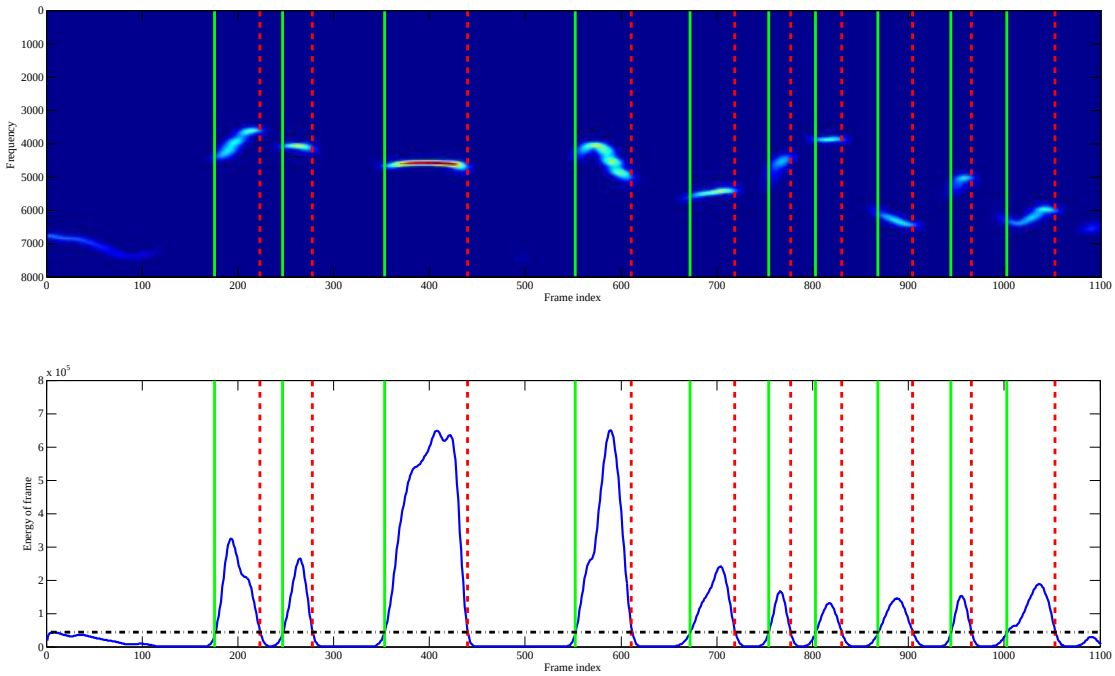


FIGURE 1.6: Energy based segmentation-1: We first compute the energy of each frame and then compute a threshold for energy. The markers for the beginning (solid green line) and end (dashed red line) of the syllable are obtained by finding the intersection of the signal with the threshold, and checking the sign of the derivative. The threshold is denoted by a black horizontal line.

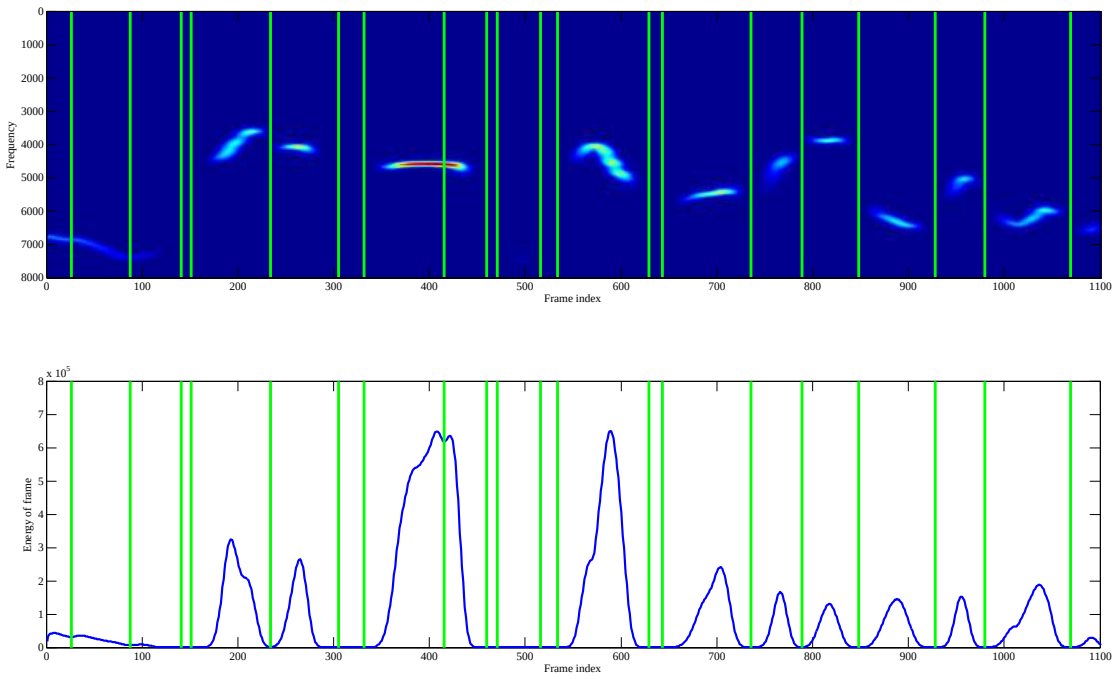


FIGURE 1.7: Energy based segmentation-2: We first compute the energy of each frame and then determine sharp local minima. The vertical markers denote potential syllable boundaries.

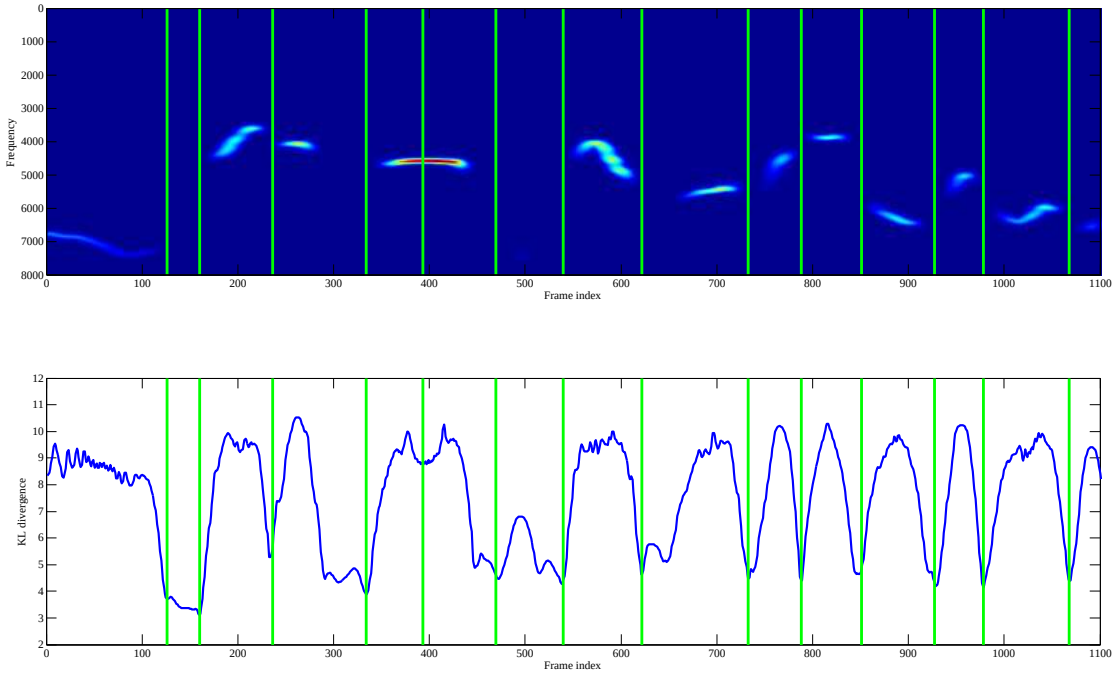


FIGURE 1.8: KL-divergence based segmentation: The KL divergence between the normalized PSD of each frame and the uniform distribution is computed. Next, we determine the sharp local minima in the KL-divergence plot. The vertical markers denote potential syllable boundaries.

tral structure (due to harmonic nature of bird vocalizations) and their constituent frames usually have high values of KL divergence while noisy frames usually have low values of KL divergence. We can use an algorithm to determine sharp local minima of the KL divergence. Fig. 1.8 provides an example of the KL-divergence based segmentation approach. Similar to the energy based segmentation, we can merge adjacent elements.

In practice, it is beneficial to use a combination of methods for segmentation. Energy based segmentation approaches are effective for bird sound recordings with

high signal-to-noise ratio (SNR). The KL-divergence based segmentation approach helps identify tonal regions in the spectrogram, but does not account for the actual energy in the frame. Hence, these approaches complement each other in their ability to identify bird syllables. Here, we use a combination of KL-divergence and energy-based algorithms for segmentation. Further details and illustrations of our segmentation algorithm are available in Appendix A.

1.2.3 Frame level features

After segmentation, the next task is to compute frame level features. We consider the following frame level features:

Spectrum PDF

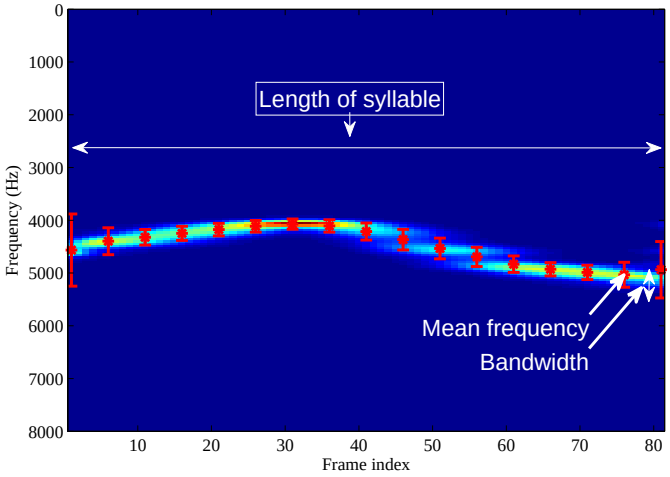
Spectrum PDF refers to the normalized PSD for each frame.

f_c (Mean frequency) and BW (Bandwidth)

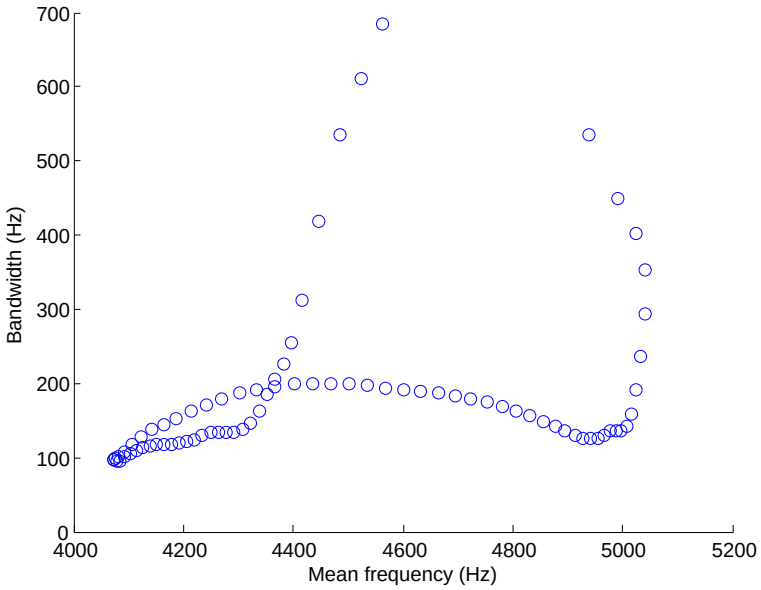
The mean frequency, f_c and the bandwidth, BW of each frame correspond to the mean and standard deviation of the filtered Spectral PDF, where the normalized PSD specifies the probability density function. Highly tonal vocalizations will have lower bandwidth. Fig. 1.9(a) illustrates the computation of mean frequency and bandwidth for each frame and Fig. 1.9(b) displays a scatter plot of the mean frequency and bandwidth of all the frames in this syllable.

Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs are a popular feature representation for audio signals and have been successfully applied for human speech recognition. MFCCs approximate the human auditory system's response by choosing a set of equally spaced frequency bands on the Mel scale as the basis function. However, they are not as simple to visualize as the other features used here. MFCCs are computed as follows: [9]



(a)



(b)

FIGURE 1.9: (a) Computation of frame level features from the spectrogram: Mean frequency, Bandwidth of a subset of frames, Length of the syllable is the number of frames within the syllable (b) Scatter plot of mean frequency and bandwidth of frames within the syllable

1. Compute the PSD of a frame.

2. Map the PSD obtained above onto the mel scale, using triangular overlapping windows.

3. Compute the logarithms of the powers at each of the mel frequencies.

4. Compute the discrete cosine transform (DCT) of the mel log powers.

5. The MFCCs are the magnitudes of the resulting DCT coefficients.

Other frame level features

Some other frame level features include the energy of the frame, zero crossing rate, KL-divergence between the PSD and uniform distribution.

1.2.4 Interval and Syllable level features

Once we obtain the frame level features, we need to obtain a feature vector representation at the syllable level or the interval level (depending on the model in Chapter 3.). We will describe the computation of syllable level features here, the interval level features can be computed in a similar fashion. The syllable level features can be computed using the following methods:

1. Compute the mean of the frame level features of the constituent frames in the syllable,

2. Parametrize the distribution of the frame level features within the syllable, and use the maximum likelihood parameter estimates as the feature vector representation of the syllable. In general, we can use distributions that capture

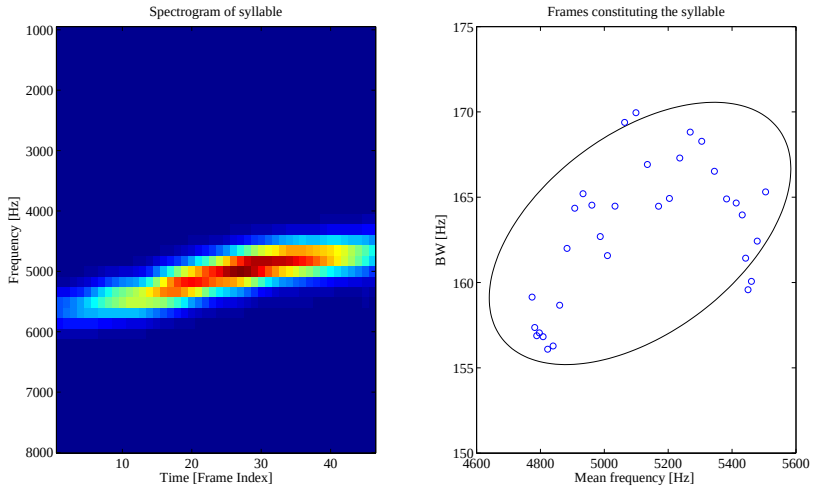


FIGURE 1.10: Example of syllable level feature vector computation: The syllable is parametrized by the parameters of the Gaussian distribution (mean, covariance) fitted to the frame level features (here, mean frequency and bandwidth).

temporal dependence between the frames and obtain features that capture temporal dependence.

Fig. 1.10 displays the spectrogram of a syllable and the scatter plot of the mean frequency and bandwidth of the frames. The syllable is parametrized by the parameters of the Gaussian distribution fitted to the frame level features. Fig. 1.11 contains the scatter plot of the syllable level features of syllables belonging to different species. The syllable level features are the mean of the frame level features within the syllable, and the frame level features were the mean frequency and bandwidth. While mean frequency and bandwidth might appear to be very simple frame level features, we observe in Fig. 1.11 that some species are reasonably separated by just aggregating these features at the syllable level.

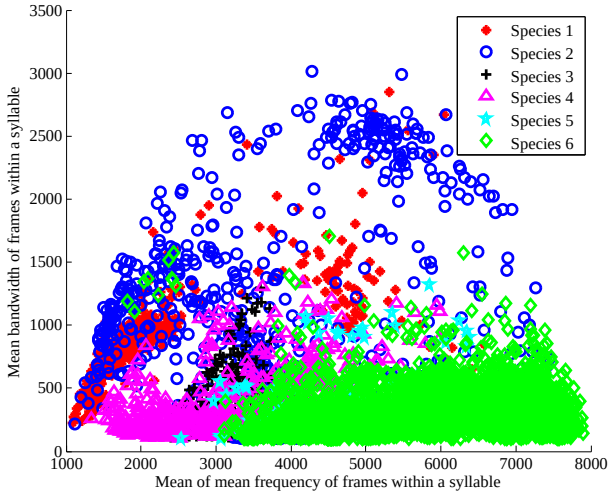
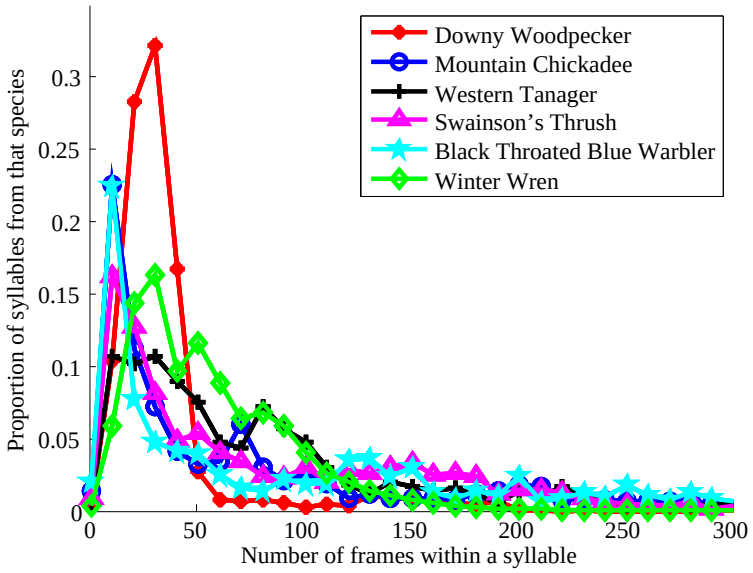


FIGURE 1.11: Example of scatter plot of syllable level features from syllables belonging to various species. In this case, the syllable level features are the mean of the mean frequency and bandwidth of the constituent frames within each syllable.

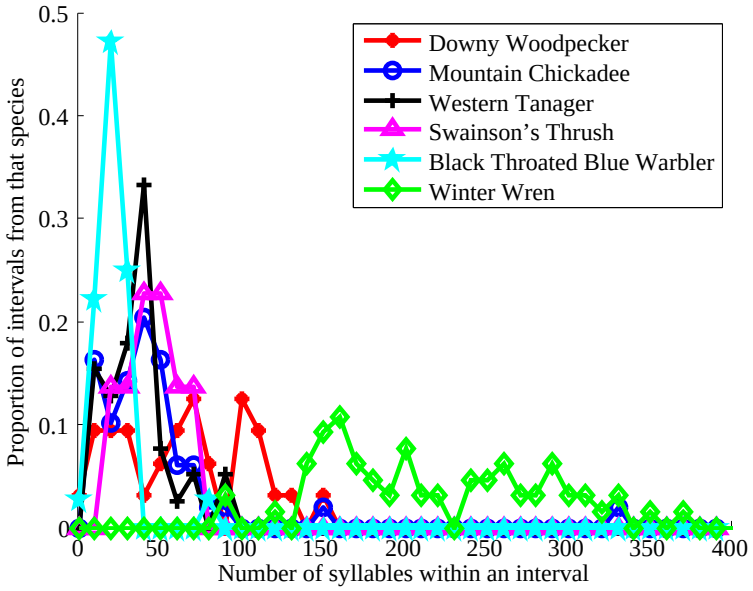
Other than the frame level features described above, one might also use features such as syllable lengths (i.e., the duration of the syllables), and the number of syllables within a recording. Fig. 1.12 shows the histogram of the syllable lengths and the number of syllables for each species. Some species such as Downy Woodpecker tend to produce a lot of short syllables, while species such as Black-throated blue warbler tend to produce few syllables. When two species vocalize in the same frequency range, but differ in their vocalization durations, these features provide valuable cues for species identification.

1.2.5 Codebooks of frame level features

In the previous section, we described several frame level features. In general, the frame level features are continuous, and they may be very high dimensional. Codebook construction allows us to map the frame level features to a discrete *word*.



(a)



(b)

FIGURE 1.12: (a) Histogram of the syllable lengths, across species. Note: Graph has been truncated for readability. Black-throated blue warbler tends to produce long syllables ($>$ 730 frames) (b) Histogram of the number of syllables per recording, across species

Fig. 1.13 illustrates the codebook construction for 2-dimensional features with a codebook containing 3 elements in the dictionary. Basically, we cluster the data points (in this case, the frame level features) and represent each data point by the index of the nearest cluster. Hence, a recording may be interpreted as a text document, where words correspond to frames, the paragraphs correspond to syllables and the document itself corresponds to the recording. The interval level histogram of the codewords in a recording allows us to obtain a bag-of-codewords representation for the recordings, similar to the popular bag-of-words representation for text documents. Codebook representations have been successfully applied for images (bag-of-visual words) as well.

1.3. Bird species identification problem

1.3.1 Problem statement

Our objective is to identify bird species based on audio recordings. In general, the training data is a collection of recordings of bird sounds, each of which is labeled with the list of species present in the recording. The recordings differ in their duration, so they are split into equal-length intervals (of duration 30 seconds). The task is to learn an acoustic model for each species based on these training set intervals so that we can correctly classify a test interval. The training dataset can be viewed as a collection of <`interval, label`> pairs, i.e., $\mathcal{S} = \{(\mathcal{D}^{(1)}, \mathbf{y}^{(1)}), (\mathcal{D}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathcal{D}^{(M)}, \mathbf{y}^{(M)})\}$, where $\mathcal{D}^{(m)}$ corresponds to the data representation of the m^{th} interval, $\mathbf{y}^{(m)}$ denotes the set of species present in the m^{th} interval and M denotes the total number of training set intervals.

We next discuss the data representation of the interval. Fig. 1.14 illustrates the

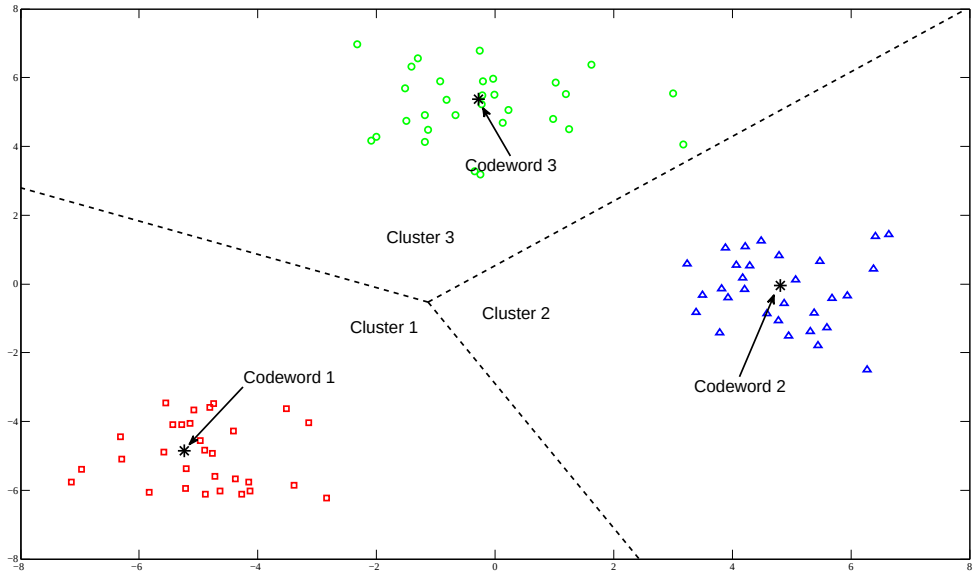


FIGURE 1.13: Illustration of the codebook construction procedure: The feature vectors are clustered and each point is represented using the index of nearest cluster center, i.e., the points in cluster 1 will be represented by 1. The dashed lines indicate the Voronoi boundaries between the clusters. Effectively, each input point is mapped to a 'codeword'.

data representation. An interval of sound can be viewed as a collection of syllables. Syllables are further divided into frames, where each frame corresponds to the sound in a very short span of time. The frames can then be represented by the frame level features described in Section 1.2.3. A syllable $\mathbf{x}(i)$ consisting of n_i frames can be viewed as a sequence of observations, i.e., syllable $\mathbf{x}(i) = [x_1(i), x_2(i), \ldots, x_{n_i}(i)]$ where observation $x_j(i)$ corresponds to the feature vector representation of the j^{th} frame in the i^{th} syllable. Mathematically, the data in an interval of sound may be represented as $\mathcal{D} = [\mathbf{x}(1), n_1, z_1, \ldots, \mathbf{x}(N), n_N, z_N, N]$, where $z_1, z_2, \ldots, z_N$ denote the syllable level labels. As an example for such a dataset, consider an extension of Fig. 1.5, where the label information is available for each of the syllables. In general, each of the syllables might belong to a different species and usually, the syllable labels are not observed. In the case where only the recording-level labels are available, the learning problem can be cast in the multiple instance multiple label (MIML) framework [10], where the bags correspond to the recordings and the instances correspond to the syllables or the frames. In the models developed in this work, we assume that each of the recordings contains just a single species, i.e., $\mathbf{y}^{(i)} \in \{1, 2, \ldots, C\}$ where C denotes the number of classes.

1.3.2 Connection to other machine learning problems

Next, we discuss the connections between the problem stated above and some of the other applications of machine learning. The bird species identification problem is similar to:

- *Image classification:* Images usually contain labels about multiple objects present in the image. It is relatively easy to obtain high level labels about presence/absence of objects in the image, but difficult to obtain fine-grained

annotation for the images.

- *Document classification:* Consider the case where the frame level feature vectors are discrete words (for instance, obtained through a codebook representation). In this case, the audio recording may be thought of as a text document. Many text documents on the web contain multiple 'tags' (or categories), but it is expensive for humans to point out which portion of the text is responsible for the tag.

A popular representation for text documents is the 'bag-of-words' representation where we represent a document by an empirical histogram computed over the dictionary, i.e., each document is represented as a vector of size equal to that of the dictionary, with each entry denoting the number of times the corresponding word occurred in that particular document. Note that the bag-of-words representation does not make use of information regarding the order of the words within the document. By using the bag-of-words representation, documents of varying lengths are transformed to a fixed length vector (of size equal to the dictionary size of the document corpus). The bag-of-words representation has been very successfully used in many text classification problems. Images may be converted to a bag-of-visual-words representation as described in Section 1.2.5 and ideas from the text classification domain can be applied for image classification.

1.3.3 Previous work in bird species identification

Bird species can be classified using features extracted from audio recordings. Different feature representations and machine learning methods have been applied for bird species identification in the literature [3, 7, 11–13]. A common approach to bird species classification is to identify distinct syllables, then construct feature

$$\overbrace{\underbrace{(D^{(1)}, \mathbf{y}^{(1)})}_{\text{Interval 1}} \quad , (D^{(2)}, \mathbf{y}^{(2)}), \ldots}^{\underbrace{\{\mathbf{x}(1)^{(1)}}_{\text{Syllable 1}} \ , n_1^{(1)}, z_1^{(1)}\}, \ldots, \{\mathbf{x}(N)^{(1)}, n_N^{(1)}, z_N^{(1)}\}, N^{(1)}}$$
$$\overbrace{[x_1(1)^{(1)}, \ldots, x_{n_1}(1)^{(1)}]}$$

FIGURE 1.14: Data representation for bird sound recordings: $\mathbf{y}^{(i)}$ denotes the set of species present in the i^{th} interval. Each interval is divided into syllables which are further divided into frames. In general, each of the syllables might belong to a different species.

vectors for those syllables and apply a standard classifier such as nearest neighbor or support vector machines to predict the species for each syllable [3, 5, 7, 8, 14–16]. In [11], the authors used dynamic time warping (DTW) to compare the input spectrograms with a predefined set of templates. In [12], the authors used neural networks and multivariate statistical techniques in conjunction with a set of temporal and spectral features. In [13], the authors used wavelet coefficients along with self organizing map (SOM) and multilayer perceptron (MLP). In [3], the authors compared three different feature representations (sinusoidal model, Mel-cepstrum model, descriptive parameters) by evaluating their performance with different classification algorithms based on DTW, Gaussian mixture model (GMM), Hidden Markov model (HMM). In [7], the author used a decision tree based classifier with support vector machine (SVM) at each node. Song-level species prediction has also been investigated using Hidden Markov Models [3, 17], Gaussian Mixture Models [3], based on comparisons of syllable-pair histograms [18], or nearest-neighbor classifiers using a feature constructed by aggregating syllable features [8].

Audio classification in general has been widely studied, with applications to human speech and music being the most common. Probabilistic topic models have also been applied for unsupervised learning of musical key profiles [19]. The interval-level model is closely related to recent work by Seyerlehner et al. [20] on music genre identification. They follow a codebook approach to constructing audio feature histograms, and use a nearest-neighbor classifier with ℓ_1 distance to classify these features. However, it is not obvious why a nearest-neighbor classifier is ideal for classifying histograms of features, or which distance measures are the best for comparing histograms. Here, we show that the Bayes optimal classifier for a probability model for audio is closely related to nearest-neighbor classifiers using histograms of features with appropriate distance measures.

1.4. Probabilistic models for bird species identification

1.4.1 Background

Probabilistic models have been successfully applied in domains such as information retrieval, computer vision, bio-informatics and speech processing. Probabilistic models allow us to encode our assumptions about the data explicitly, while simultaneously enabling us to perform conventional machine learning tasks such as classification and clustering. Probabilistic graphical models can be used to capture the hierarchical nature of the manner in which the data is generated. For example, hierarchical Bayesian models can be used to model the distributions at syllable level as well as interval level in Fig. 1.14. The desired properties of the parameters (eg., sparsity) may be favored by choosing appropriate prior distributions for the parameters. In addition, probabilistic models are flexible, allowing simultaneous modeling

of various types of information, for instance, the models under consideration can be readily extended to handle multiple-labels and additional information such as annotations or tags.

Recently, there has been a lot of work in the development of probabilistic models for text documents. Particularly, an interesting problem is that of learning the 'topics' present in a document corpus. The basic assumption is that each document contains a few topics, where topics may be interpreted as clusters of words. Intuitively, these methods find a lower dimensional representation of the documents through a low-rank approximation to the original bag-of-words representation, i.e.,

$$\texttt{Vocabulary} \times \texttt{Documents} \;\approx\; (\texttt{Vocabulary} \times \texttt{Topics}) \cdot (\texttt{Topics} \times \texttt{Documents})$$

The models differ in the way they characterize the quality of the approximation (eg., ℓ_2 loss, KL divergence) and their assumptions about the prior distributions about the low rank factors. Latent Semantic Indexing (LSI) computes a lower dimensional representation of the bag-of-words representation by computing a singular value decomposition. Probabilistic Latent Semantic Analysis (PLSA) [21] introduced a probabilistic generative model for the documents and employed Expectation Maximization (EM) algorithm for parameter estimation. Another popular approach is Non-Negative Matrix Factorization (NNMF), where the low rank factors are constrained to be non-negative [22]. Latent Dirichlet Allocation (LDA) [23] introduced a Bayesian version of the PLSA model. A wide variety of inference algorithms have been developed for inference in LDA [23–26]. However, LDA is an unsupervised model and cannot be directly used for classification. Supervised extensions of LDA have been proposed for applications such as image classification [27–29], document classification [30–32], movie rating prediction [33], named entity mining [32, 34],

credit attribution in multi-labeled corpora [35].

1.4.2 Motivation for new models and inference methods

In this thesis, we develop efficient inference techniques for existing models, as well as develop probabilistic models tailored to bioacoustic data. First, we develop inference algorithms for the supervised latent Dirichlet allocation (LDA) model introduced in [27]. While the effect of inference algorithm has been studied for unsupervised LDA [26], it is not obvious which inference algorithm is suitable for supervised LDA. Supervised LDA models are evaluated based on classification accuracy and moreover, the inference for test data is different compared to the unsupervised LDA model. Unlike text documents where the dictionary size is large and there are relatively few words per document, images and audio contain large number of words per document but the dictionary size (i.e., the codebook size) is relatively small. Clearly, there is a need for efficient inference techniques for parameter estimation and classification in supervised LDA models. computational complexity and classification performance of the inference methods for supervised LDA. We evaluate our results on the bird species identification task, as well as, image classification and document classification tasks.

Next, we present novel probabilistic models for bird sound recordings that can account for the temporal structure at the block level (blocks may correspond to syllables or the entire interval), as well as model additional information such as the length of the vocalizations and the frequency of vocalizations. Such information is not captured by existing models, and we have shown that these can help in species identification. The frame level features considered here can take on continuous values unlike the discrete codewords in Chapter 2. We use a non-parametric density

estimation procedure and show that the MAP classifier can be interpreted as a nearest neighbor classifier with an appropriate distance criterion. We evaluate the classification accuracy obtained by our models on the species identification task and compare their performance to an SVM-based approach.

1.4.3 Organization of this thesis

In Chapter 2., we discuss the inference algorithms for supervised LDA model. In Chapter 3., we discuss the probabilistic models tailored to bird sound recordings. Finally, in Chapter 4., we provide a summary and discuss future work. For readability, we have made the chapters self-contained, and the notation correspond to [6, 36].

2. SUPERVISED LATENT DIRICHLET ALLOCATION

2.1. Background on topic models

Latent Dirichlet allocation (LDA) [23] is an unsupervised latent variable model originally applied in the field of document modeling due to its ability to decompose documents into topics and uncover topics decomposition into words in a concise manner. As an unsupervised model, LDA can be used to perform dimensionality reduction by mapping the high dimensional bag-of-words representation to lower dimensional topic representation.

Recently, there has been a growing interest in supervised extensions of LDA for applications such as image classification [27–29], document classification [30–32], movie rating prediction [33], named entity mining [32,34], credit attribution in multi-labeled corpora [35]. In this paper, we focus on the supervised LDA model introduced in [27]. The motivations for supervised LDA are multi fold. Supervised LDA can help in identifying topics specific to a particular class. In addition, probabilistic models are flexible, allowing simultaneous modeling of various types of information, for instance, the supervised LDA model under consideration can be readily extended to handle multiple-labels and additional information such as annotations or tags.

Despite the ability of topic models to produce a concise representation, parameter estimation in topic models remains a challenging task. In most cases, exact inference is intractable and hence, approximate inference methods are required. Inference methods for topic models can be broadly categorized into sampling based approaches and deterministic approximations. Recent work stresses the importance

of properly adapting the priors (hyperparameters) in LDA-based models [26, 37]. This can be addressed by optimizing the prior parameterization instead of using a fixed prior. An excellent comparison of various inference algorithms for LDA such as variational Bayes (VB), collapsed Gibbs sampling (CGS), collapsed variational Bayes (CVB) and maximum a-posteriori (MAP) inference, is available in [26].

Previous work in the supervised LDA model employed VB for inference [27]. We derive the MAP, CGS and CVB inference solutions for supervised LDA and study the effect of the choice of inference algorithm for supervised LDA. While the extension of [26] to the supervised case might appear straightforward at first sight, several new aspects arise in the supervised case:

- *Model based classification:* The classification stage is completely new relative to LDA and requires the development of efficient inference techniques for classification of test documents.

- *Classification accuracy:* Supervised LDA is evaluated in terms of classification accuracy rather than perplexity. It is not obvious which inference algorithm leads to the best classification accuracy.

- *Train vs Test computational complexity:* While the training complexity of supervised LDA is similar to that of LDA, model based classification approach for Supervised LDA requires significantly additional computation in the test stage than LDA, and can be computationally intensive when the number of classes is large. We introduce a new classification approach to solve this problem.

In this work, we address the following questions:

1. Which inference algorithm provides a good trade-off between classification accuracy and computational complexity for Supervised LDA?

2. Which inference algorithm is suitable when using LDA as just an unsupervised pre-processing technique (i.e., when the topic representation is used as an input to a generic classifier)?

2.2. Problem statement

The training data is assumed to be a collection of M documents along with their corresponding labels. The collection of N_i words for the i^{th} document is denoted by $\mathbf{w}_i = \{w_{i1}, \ldots, w_{iN_i}\}$ and the label associated with the i^{th} document is denoted by y_i. The entire corpus can then be represented by $(\mathbf{W}, \mathbf{Y})$ where $\mathbf{Y} = (y_1, y_2, \ldots, y_M)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M)$. We assume that each document belongs to one of C classes, i.e., $y_i \in \{1, 2, \ldots, C\}$. We refer the reader to Table 2.1 for an explanation of the symbols used. The task is to learn a model for $(\mathbf{W}, \mathbf{Y})$ so that we are able to classify a new test document $\mathbf{w}_t$. Next, we discuss the details of the supervised LDA model used in this paper.

2.3. Generative process for supervised LDA

Supervised LDA [27] is a natural extension to the original LDA model [23]. The graphical model for the supervised LDA is shown in Fig. 2.1 and the generative process is explained in Algorithm 1.

The key difference between supervised LDA and LDA is that, for each training

Algorithm 1 Generative process

for $k = 1$ to K **do**

Draw $\phi_{k,\cdot} \sim \text{Dirichlet}(\boldsymbol{\beta})$

end for

for $i = 1$ to M **do**

Draw y_i, N_i

Draw $\theta_i \sim \text{Dirichlet}(\alpha_{y_i,\cdot})$

for $j = 1$ to N_i **do**

Draw $z_{ij} \sim \text{Discrete}(\theta_i)$

Draw $w_{ij} \sim \text{Discrete}(\phi_{z_{ij},\cdot})$

end for

end for

document, we first draw the label y and then choose a class-dependent Dirichlet prior for the topic proportions. The Dirichlet prior over topics is represented as a $C \times K$ matrix $\boldsymbol{\alpha}$ where c^{th} row of $\boldsymbol{\alpha}$ matrix corresponds to the Dirichlet prior for class c. Note that we consider both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to be asymmetric Dirichlet priors. The number of words in each document, N_i, is an ancillary variable and we assume that it is independent of the class c. The supervised LDA model may also be viewed as a special case of models such as Labeled-LDA model [35] and Dirichlet-Multinomial Regression model [38].

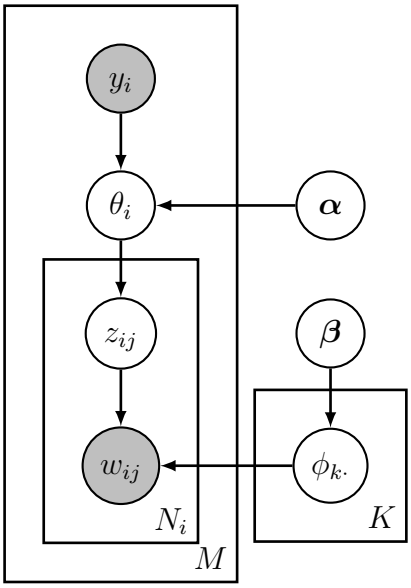


FIGURE 2.1: Graphical model for Supervised LDA. The shaded nodes correspond to the observations.

2.4. Parameter estimation in supervised LDA

Parameter estimation in Supervised LDA is based on the maximum marginal likelihood principle. The marginal likelihood of the data i.e., the likelihood of $(\mathbf{W}, \mathbf{Y})$ conditioned on the hyperparameters, is given by

$$p(\mathbf{W}, \mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\phi}} \sum_{\mathbf{Z}} p(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \; d\boldsymbol{\theta} d\boldsymbol{\phi},$$

where $\mathbf{Z}$ corresponds to the topic assignments of all the words in the training corpus. The above integral is intractable. Deterministic approaches (such as VB) replace the integral with a tractable lower bound. Sampling based approaches (such as CGS) approximate this integral (expectation) using an empirical (sample-based) average. The MAP estimation procedure approximates the integral by using point estimates

TABLE 2.1: List of symbols

C	Number of classes
M	Number of training documents
N_i	Number of words in i^{th} training document
K	Number of topics
V	Vocabulary size
$\mathbf{W}$	All the words in the training documents
$\mathbf{w}_i$	$1 \times N_i$ vector containing the words in document i
$\mathbf{z}_i$	$1 \times N_i$ vector containing topic assignments of corresponding words in $\mathbf{w}_i$
$\mathbf{Y}$	Labels of the training documents
$\boldsymbol{\theta}$	$K \times M$ matrix whose i^{th} column represents the 'topic-multinomial' parameter for the i^{th} training document
$\boldsymbol{\phi}$	$K \times V$ matrix where ϕ_{kl} denotes the probability of l^{th} word given k^{th} topic
$\boldsymbol{\alpha}$	$C \times K$ matrix where $\alpha_{c,\cdot}$ denotes the Dirichlet prior for class c
$\boldsymbol{\beta}$	$1 \times V$ vector which denotes the Dirichlet prior for each row of $\boldsymbol{\phi}$

of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ ($\mathbf{Z}$ can be marginalized out).

To the best of our knowledge, only VB inference has been explored earlier for the supervised LDA model [27]. We derive the update equations for MAP, CGS and CVB0 for the supervised LDA model. Not surprisingly, if we set all the $\mathbf{Y}$ to be equal in our update equations, we recover the update equations for the unsupervised LDA case.

2.4.1 MAP estimation

The MAP estimate of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is given by

$$\boldsymbol{\theta}^*, \boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \; p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{2.1}$$

As shown in Appendix B, the objective function for MAP is given by

$$\begin{aligned} \log p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \;\; = \;\; & \sum_i \Big(\sum_{v=1}^{V} \Big[n_{vi} \log (\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi} \Big] + \log P(y_i) \\ & + \sum_{k=1}^{K} (\alpha_{y_i,k} - 1) \log \theta_{ik} - \log B(\alpha_{y_i,\cdot}) \Big) \\ & + \sum_k \Big(\sum_v (\beta_v - 1) \log \phi_{kv} - \log B(\boldsymbol{\beta}) \Big), \end{aligned} \tag{2.2}$$

where $B(\cdot)$ denotes the multinomial beta function. Parameter estimation is performed by maximizing (2.2) w.r.t. $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ in a coordinate ascent fashion. As shown in Appendix D, the updates for θ_{ik} and ϕ_{kl} are given by,

$$\theta_{ik} \propto \max(g_{ik}, 0), \qquad \phi_{kv} \propto \max(h_{kv}, 0) \tag{2.3}$$

where $\sum_k \theta_{ik} = 1$, $\sum_v \phi_{kv} = 1$ and

$$\begin{aligned} g_{ik} \;\; &= \;\; \sum_{v=1}^{V} \Big[n_{vi} \frac{\phi_{kv} \hat{\theta}_{ik}}{\sum_{k'=1}^{K} \phi_{k'v} \hat{\theta}_{ik'}} \Big] + (\alpha_{y_i,k} - 1), \\ h_{kv} \;\; &= \;\; \sum_i \Big[n_{vi} \frac{\theta_{ik} \hat{\phi}_{kv}}{\sum_{k'=1}^{K} \theta_{ik'} \hat{\phi}_{k'v}} \Big] + (\beta_v - 1). \end{aligned} \tag{2.4}$$

Note that $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}$ denote the values of $\boldsymbol{\theta}, \boldsymbol{\phi}$ from the previous iteration.

Connection to NNMF

Note that (2.2) resembles the objective function of KL-divergence minimizing non-negative matrix factorization (NNMF) [39], where we have additional regularization terms on $\boldsymbol{\theta}, \boldsymbol{\phi}$. The equivalence between EM updates for PLSA and KL-divergence

minimizing NNMF updates in the unregularized case (i.e., $\boldsymbol{\alpha} = 1, \boldsymbol{\beta} = 1$ in (2.2)), has been observed in [40]. In the EM algorithm for MAP solution in LDA, the E-step involves the computation of $\gamma_{wjk} = P(z_{ij} = k|w_{ij}, \theta_i)$ and the M-step involves maximization w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. To ensure that γ_{wjk}'s are valid probabilities, [26] impose the constraint $\boldsymbol{\alpha} > 1, \boldsymbol{\beta} > 1$ in their MAP solution. Even if $\boldsymbol{\alpha} < 1, \boldsymbol{\beta} < 1$, γ_{wjk} can be valid probabilities if $g_{ik} \geq 0$ and $h_{kl} \geq 0$ in (2.4). Another subtle difference exists. In (2.2), the hyperparameters are optimized using Maximum likelihood (ML) estimation for Dirichlet distribution whereas in the MAP solution by [26], the hyperparameters are optimized using ML for Polya distribution. The Polya distribution accounts for the number of words in the document (and hence the number of topic variables), whereas the Dirichlet distribution estimates $\boldsymbol{\alpha}$ using the $\theta_i's$ and hence, does not account for document length in the hyperparameter estimation.

2.4.2 Collapsed Gibbs sampling (CGS)

In this section, we present the collapsed Gibbs sampling updates for supervised LDA. They can be derived in a similar fashion as CGS for LDA [24, 26].

$$P(z_{ij} = k|w_{ij} = v, \mathbf{Z}^{\setminus ij}, y_i) \propto (n_{ki}^{\setminus ij} + \alpha_{y_i,k}) \frac{n_{kv}^{\setminus ij} + \beta_v}{n_k^{\setminus ij} + \mathbf{1}^\top \boldsymbol{\beta}},$$

where $n_{kv}^{\setminus ij} = \sum_{(i',j') \notin (i,j)} \mathbb{1}[z_{i'j'} = k, w_{i'j'} = v]$, $n_k^{\setminus ij} = \sum_{(i',j') \notin (i,j)} \mathbb{1}[z_{i'j'} = k]$ and $n_{ki}^{\setminus ij} = \sum_{j' \neq j} \mathbb{1}[z_{ij'} = k]$. Note that $\setminus ij$ indicates that the current word-topic pair has been excluded from the counts.

2.4.3 Collapsed Variational Bayes (CVB0)

In this section, we present the collapsed variational updates for supervised LDA. As in LDA, the variational distribution is assumed to factorize as follows

[25, 26]:

$$q(\mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{Z}) \prod_i \prod_j q(z_{ij}), \tag{2.5}$$

where $q(z_{ij})$ is a multinomial distribution with parameters given by $q(z_{ij} = k) = \gamma_{ijk}$. Using the zeroth order approximation to the variational bound (which we will refer to as CVB0), the updates for γ_{ijk} are obtained as

$$\gamma_{ijk} \;\propto\; (n_{ki}^{\backslash ij} + \alpha_{y_i,k}) \frac{n_{kv}^{\backslash ij} + \beta_v}{n_k^{\backslash ij} + \mathbf{1}^\top \boldsymbol{\beta}} \tag{2.6}$$

where $n_{kv}^{\backslash ij} = \sum_{(i',j') \notin (i,j), w_{i'j'} = v} \gamma_{i'j'k}$, $n_{ki}^{\backslash ij} = \sum_{j' \neq j} \gamma_{ij'k}$, and $n_k^{\backslash ij} = \sum_{(i',j') \notin (i,j)} \gamma_{i'j'k}$.

For both CVB0 and CGS, upon convergence, the parameter estimates are computed using $\hat{\phi}_{kv} \propto (n_{kv} + \beta_v)$, where $\sum_v \hat{\phi}_{kv} = 1$. Note that while MAP is inherently parallelizable i.e., the θ_i's for the documents can be updated in parallel, the collapsed inference algorithms (CGS and CVB0) are not inherently parallelizable.

2.5. Classification

For a test document $\mathbf{w}_t$, model-based classification is made using the MAP rule, i.e.,

$$y_t^* = \arg\max_{y_t} \; p(y_t|\mathbf{w}_t) = \arg\max_{y_t} \; p(y_t, \mathbf{w}_t). \tag{2.7}$$

2.5.1 Classification using VB

VB can be used to classify a test document as follows [27]

$$y_t^* = \arg\max_{c} \; p(\mathbf{w}_t, y_t = c).$$

Since the RHS is intractable, they compare the variational lower bounds for $\log p(\mathbf{w}_t, y_t = c)$. The variational lower bound may be computed as follows [41]

$$\log p(\mathbf{w}_t, y_t = c) \geq E_q[\log p(\mathbf{w}_t, y_t = c)] + H[q], \tag{2.8}$$

where q denotes the fully factorized variational distribution $q(\theta_t, \mathbf{z}_t) = q(\theta_t) \prod_{j=1}^{N_t} q(z_{tj})$. This approach requires recomputation of the variational lower bound for each possible value of c, and can be computationally demanding when C is large.

Next, we present the classification rules for MAP and CVB0.

2.5.2 Classification using MAP

We express $p(y_t|\mathbf{w}_t)$ as follows

$$y_t^* = \arg\max_{y_t} \int_{\theta_t} p(\theta_t|\mathbf{w}_t) p(y_t|\theta_t)\ d\theta_t. \tag{2.9}$$

Since the integral in (2.9) cannot be computed in closed-form, we propose to approximate it as follows

$$\begin{aligned} y_t^* &\approx \arg\max_{y_t}\ p(y_t|\theta_t^*) \int_{\theta_t} p(\theta_t|\mathbf{w}_t)\ d\theta_t & (2.10)\\ &= \arg\max_{y_t}\ p(y_t|\theta_t^*), & (2.11) \end{aligned}$$

where,

$$\begin{aligned} \theta_t^* &= \arg\max_{\theta_t}\ p(\theta_t|\mathbf{w}_t), & (2.12)\\ &= \arg\max_{\theta_t}\ \log p(\mathbf{w}_t|\theta_t) + \log p(\theta_t). & (2.13) \end{aligned}$$

The approximation in (2.10) may be interpreted as a zeroth order version of Laplace approximation of the integral in (2.9) around θ_t^*. Additionally, obtaining a single θ_t^* (rather than C) enables us to think of supervised LDA as a supervised dimensionality

reduction method. Note that θ_t^* can be obtained by using an optimization similar to (2.2). Since y is unobserved for test data, we have $\log p(\theta_t)$ which is a mixture of Dirichlet distributions, instead of $\log p(\theta_i|y_i)$ used in training (2.2). We treat y_t as a latent variable and derive an EM algorithm to compute θ_t^*. The update rule is given by $\theta_{tk} \propto \max(g_{tk}, 0)$, where $\sum_k \theta_{tk} = 1$, and

$$g_{tk} = \sum_{l=1}^{V} \Big[n_{lt} \frac{\phi_{kl} \hat{\theta}_t(k)}{\sum_{u=1}^{K} \phi_{ul} \hat{\theta}_t(u)} \Big] + \sum_{c=1}^{C} P(y_t = c|\hat{\theta}_t)(\alpha_{ck} - 1).$$

Note that $\hat{\theta}_t$ denotes the value from the previous iteration. Note the similarity of (2.14) to (2.4). Since y_t is not observed, the $(\alpha_{ck}-1)$ term is weighted by $P(y_t = c|\hat{\theta}_t)$.

2.5.3 Classification using CVB0

We consider two classification rules for CVB0. First, we classify y_t^* using

$$y_t^* = \arg\max_c \; p(\mathbf{w}_t, y_t = c),$$

Since the RHS is intractable, we use the collapsed variational lower bound for $\log p(\mathbf{w}_t, y_t = c)$. The collapsed variational lower bound is computed using (2.8), where the expectation is w.r.t the collapsed variational distribution, given by $q(\theta_t, \mathbf{z}_t) = q(\theta_t|\mathbf{z}_t) \prod_{j=1}^{N_t} q(z_{tj})$. We refer to this classifier as CVB0-1. Note that this approach can be computationally intensive when C is large. We introduce a second approach to alleviate this problem,

$$\gamma_{tjk} \;\propto\; (n_{kt}^{\backslash ij} + \sum_{c=1}^{C} P(y_t = c|\hat{n}_{\cdot t}) \; \alpha_{c,k}) \; \hat{\phi}_{kv} \tag{2.14}$$

where $n_{kt}^{\backslash tj} = \sum_{j' \neq j} \gamma_{tj'k}$, $\hat{n}_{\cdot t}$ denotes the value of $n_{\cdot t}$ from the previous iteration and $\hat{\phi}$ denotes the estimate of ϕ computed from the training data. We will refer to the second approximation as CVB0-2.

2.6. Experimental Results

In the first experiment, we compare the classification accuracy and runtime achieved by MAP, CVB0, VB and PLSA for the supervised LDA model. In the second experiment, we compare the classification accuracy achieved by MAP, CVB0 when LDA is employed as a dimensionality reduction method. We implemented CGS and observed that the runtime associated with CGS is significantly larger than the runtime associated with the other methods. In CGS, we need to compute the topic probabilities for each occurrence of a word (and not just every unique occurrence as in the other methods) as well as draw multiple topic samples before estimating the hyperparameters. Hence, we do not include the results obtained using CGS here.

2.6.1 Implementation details

All the methods were implemented in Matlab. We used similar vectorization techniques in all of our implementations. We plan to make our code publicly available in the near future.

Hyperparameter optimization for MAP

Species identification, Image classification: We restricted $\boldsymbol{\alpha} \geq 1$, but did not impose any constraint on $\boldsymbol{\beta}$. In our experiments, we used a log-barrier method with Newton update equations [42] to compute the optimum $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ in (2.2).

Document classification: Since the documents are of significantly varying lengths, we computed $n_{ki} = N_i\theta_{ki}$ and estimated $\boldsymbol{\alpha}$ using unconstrained ML for Polya distribution. Similarly, we compute n_{kv} and estimate $\boldsymbol{\beta}$.

Hyperparameter optimization for CVB0

We used the fixed point updates in [43] to compute the ML estimates of Polya dis-

tribution.

Hyperparameter optimization for VB

As done by [27], we optimize $\boldsymbol{\alpha}$, but set $\boldsymbol{\beta}$ to 1.

PLSA-NN

As observed by [44], PLSA is equivalent to LDA when all the hyperparameters are set to 1. Hence, we used the same update equations as that of the MAP estimation, except that the hyperparameters are held constant at 1. Once the θ's have been obtained for the training as well as the test data, we use a k-nearest neighbor (k-NN) classifier with Euclidean distance metric for classifying the test documents. Following [45], we set the number of nearest neighbors, $k = 10$.

2.6.2 Datasets

In this section, we describe the datasets used in our experiments.

Species identification (Audio classification)

[1] We used recordings from the Cornell Macaulay library, of 6 species: Downy Woodpecker, Mountain Chickadee, Western Tanager, Swainson's Thrush, Black Throated Blue Warbler, and Winter Wren. The recordings were collected over several decades, mainly in the western United States. Most are made using a directional microphone in the field. The amount of noise in the recordings varies widely. In addition to static and wind, some recordings contain cars sounds, human speech, and other non-bird sounds. We manually removed most portions of sound with human voices. Although each recording is labeled with just one species, some recordings contain multiple birds, sometimes of different species; usually the loudest bird present cor-

[1] This experiment was not reported in [36].

responds to the label for the recording. The sampling frequency for all recordings is 44.1 kHz. The audio data is stored as mono-channel WAV files. All of these recordings are at least 30 seconds long, and most are less than 10 minutes. We divide each recording into intervals of 30 seconds, resulting in 265 intervals. We first apply the segmentation algorithm described in Appendix A, then compute three different frame level features described in Section 1.2.3. Next, we construct a codebook for each of the frame level features [6] to obtain the bag-of-words representation for each audio recording. For constructing codebooks, we apply the k-means++ clustering algorithm [46] to a random subset of the frame-level features from the training data set. Further details regarding our implementation are available in [6].

Image classification

LabelMe: The first image classification dataset was used in [28]. The dataset consists of 1600 images from the *LabelMe* toolbox. There are totally eight classes. For each image, the SIFT vectors are obtained and then the SIFT vectors are clustered to obtain the codebook (size=158) representation. Each image contains 2401 SIFT vectors. The pre-processed dataset in bag-of-words format was made publicly available by [28]. More details regarding the dataset are available in [28]. We used three random train-test data splits for cross validation, each time dividing the data into 800 training documents and 800 test documents.

MSRC-v2: We also evaluate our algorithms on a subset of the MSRC-v2 dataset. We used images belonging to eight groups, namely, `'book'`, `'grass, cow'`, `'tree, grass, sky'`, `'bike, building'`, `'sign'`, `'water, boat'`, `'aeroplane, grass, sky'`, `'road, building'` resulting in a total of 240 images. We divided each image into 8×8 blocks and we cluster the blocks using k-means algorithm to create a codebook of size 160. Using this codebook, we create the bag-of-words representation for

each image. Again, we use 50% of the dataset for training and 50% of the dataset for testing. While we realize that it might be possible to use more sophisticated features, our goal here is to compare inference algorithms for classification rather than find good feature vector representations. Note that quite some overlap exists between the classes themselves.

Document classification

We used the *20Newsgroups* dataset [2] for document classification. The original dataset[2] consists of 18824 documents belonging to 20 different topics (class label). We grouped the 20 topics into four classes, namely 'Computer' (`comp.`), 'Talk' (`talk.`), 'Science' (`sci.`), 'Recreational' (`rec.`). We removed words occurring more than 5000 times in the entire corpus, as well as words occurring less than 200 times in the entire corpus. This resulted in 2144 words. We retained only documents that contained at least 100 words in them. We used three random train-test data splits for crossvalidation, each time using 1990 documents in the training set and 1327 documents in the test set. In all of our experiments for this dataset, we vary number of topics as follows, $K = 20$, $K = 50$ and $K = 80$.

2.6.3 Simulation details

Image classification, Document classification

We vary the number of topics and report the classification accuracy and runtime in each case. For each train-test data split, we try three random initializations and report the best classification accuracy and the total runtime. The total runtime is

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/

the sum of the runtimes for each random initialization, which is the sum of the time for training and the time for testing. We compute the mean and standard deviation of the results based on the 3 random train-test splits. The error bars in our graphs denote the variation amongst 3 random train-test splits for cross-validation.

Species identification

We vary the number of topics from 10 to 50 in steps of 10 and report the best classification accuracy. We used the 'individual-independent' crossvalidation setup described in Section 3.7.4.

In all the above cases, we train the model till the fractional change in log likelihood, given by $abs[(ll_{new} - ll_{old})/ll_{new}]$, is less than a threshold (10^{-6} in our experiments), with an additional limit on the maximum number of iterations (300 in our experiments).

2.6.4 Effect of the inference algorithm on classification accuracy

The classification accuracy comparison is shown in Fig. 2.2, Fig. 2.3 and Table 2.2 for the LabelMe, MSRC-v2 and Newsgroup datasets respectively. We can observe that MAP provides comparable performance to VB, CVB0 in terms of classification accuracy in the LabelMe dataset. However, in the newsgroup and MSRC-v2 datasets, CVB0 outperforms MAP and performs quite similar to VB. The CVB0-2 classifier provides comparable performance to CVB0-1 classifier. The classification accuracy comparison is shown in Table. 2.3 for the Cornell Macaulay library dataset. We observe that VB consistently achieves good performance. For codebook of MFCCs, both MAP and PLSA-NN outperform VB. In general, MAP tends to perform better with fewer number of topics (always 10 here).

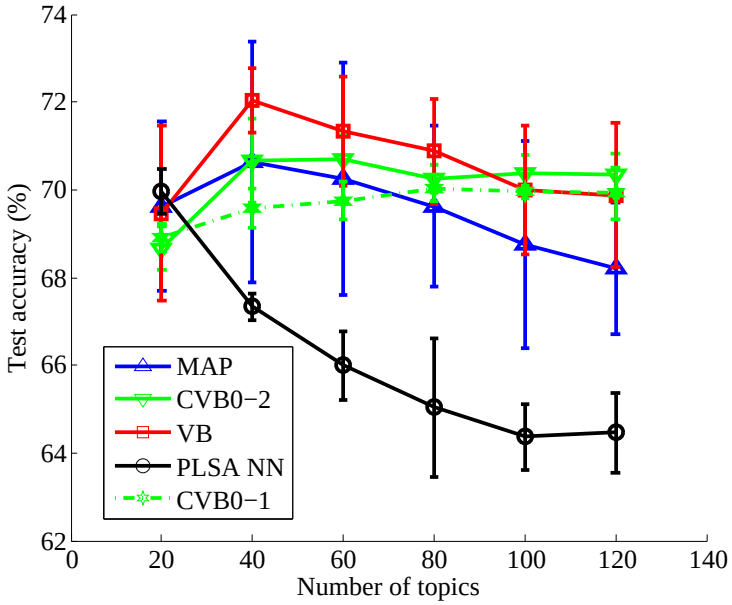


FIGURE 2.2: *LabelMe dataset:* Comparison of classification accuracy obtained using MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

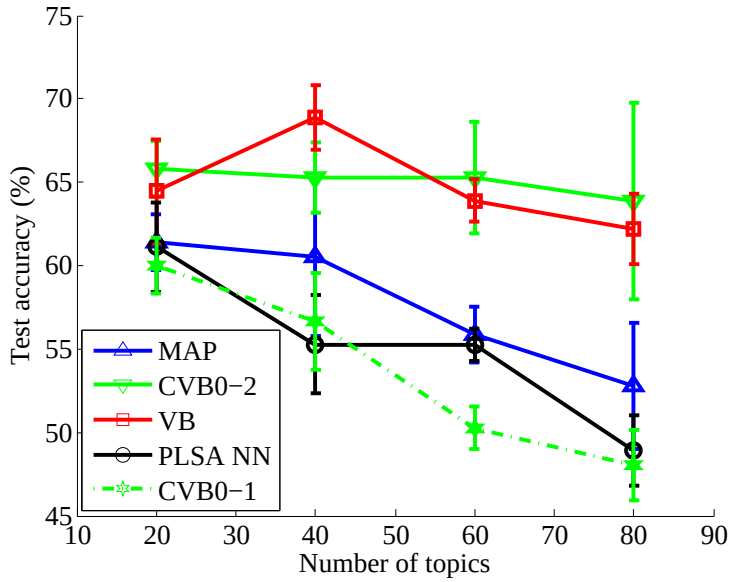


FIGURE 2.3: *MSRC-v2 dataset:* Comparison of classification accuracy obtained using MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

TABLE 2.2: *Newsgroup dataset:* Comparison of classification accuracy obtained using MAP, CVB0

Number of topics	**MAP**	**CVB0**
20	72.29 ± 0.04	85.45 ± 0.27
50	76.61 ± 1.35	83.49 ± 1.06
80	78.27 ± 1.09	81.31 ± 2.10

2.6.5 Effect of the inference algorithm while using LDA as pre-processing step

LDA can be viewed as a dimensionality reduction method. The topic representations for each document can be used as input to a discriminative classifier such as support vector machine (SVM). In this experiment, we compare the effectiveness of various inference methods, when LDA is used as a pre-processing step. To this extent, we obtained the topic representation for each document using the unsupervised LDA model and trained a SVM using the topic representation for the documents as the feature vectors. We then compare the accuracy of SVM classifier using the CVB0 and MAP feature representations. We used the Matlab interface to LibSVM [47] in our SVM experiments. We used Gaussian kernel and one-vs-one SVM. The kernel bandwidth γ and regularization parameter C_r were optimized using grid search ($[10^{-0.5}, 10^{0}, 10^{1}]$ for γ and $[10^{-2}, 10^{-1}, 10^{0}]$ for C_r).

One might also use the topic representations obtained using the Laplace approximation in Supervised LDA as input to a classifier. We compare the classifi-

TABLE 2.3: *Cornell Macaulay library dataset:* Comparison of classification accuracy obtained using MAP, CVB0, VB for codebooks of different frame level features. The optimal number of topics for each combination is included within parentheses.

Method	f_c, BW	**MFCCs**	**Spectral PDF**
MAP	81.51 (10)	**89.06 (10)**	80.38 (10)
CVB0-1	84.91 (40)	84.15 (50)	80.38 (20)
CVB0-2	84.15 (10)	86.42 (20)	82.26 (20)
PLSA-NN	81.51 (20)	88.68 (10)	85.28 (30)
VB	**88.30 (20)**	87.17 (50)	**87.92 (50)**

cation accuracy achieved by the various feature vector representations. The results are shown in Fig. 2.4 and Fig. 2.5. Surprisingly, the overall classification accuracy seems to be higher when using topic representations from unsupervised LDA.

2.6.6 Computational complexity

The runtime comparison for the LabelMe and MSRC-v2 datasets are shown in Fig. 2.6 and Fig. 2.7 respectively. We observe that MAP provides considerable advantage in terms of runtime, while providing slightly worse classification accuracy in the LabelMe dataset and significantly worse results in the MSRC-v2 dataset. The CVB0-2 classifier is significantly faster than CVB0-1. The computational complexity in the training stage for the supervised LDA is similar to the discussion in [26]. During the classification stage, CVB0-2 and MAP are roughly $O(C)$ times faster than VB and CVB0-1.

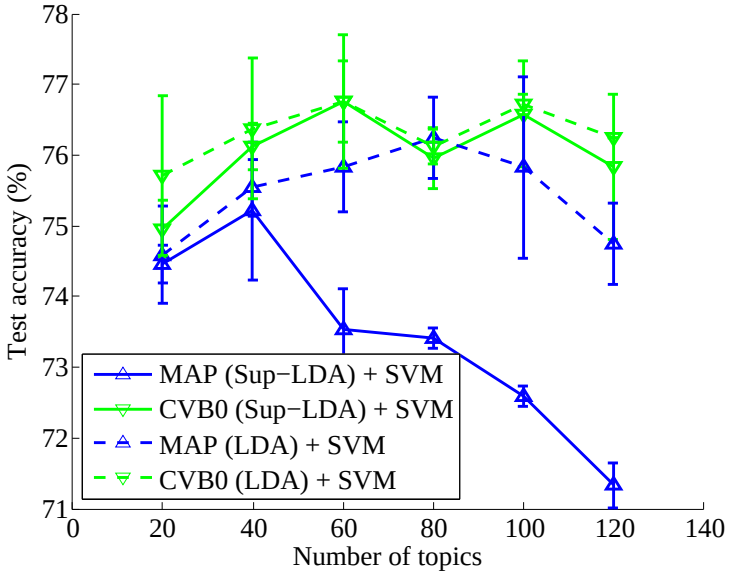


FIGURE 2.4: *LabelMe dataset:* Comparison of classification accuracy achieved by using topic representations computed using MAP, CVB0-2, as feature vector for SVM.

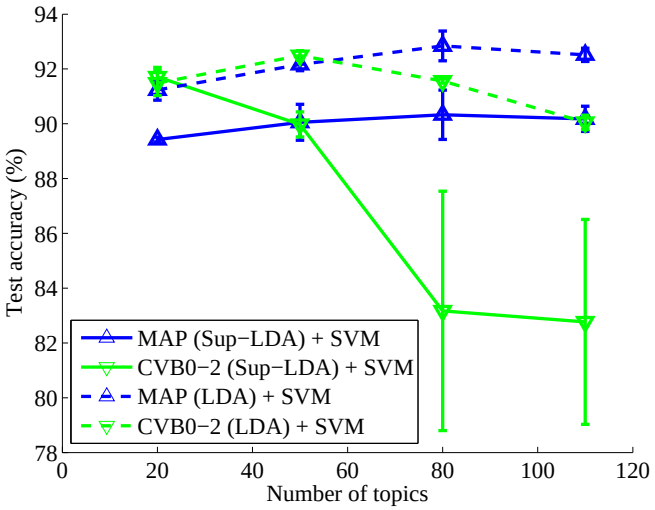


FIGURE 2.5: *4 newsgroup dataset:* Comparison of classification accuracy achieved by using topic representations computed using MAP, CVB0-2 as feature vector for SVM.

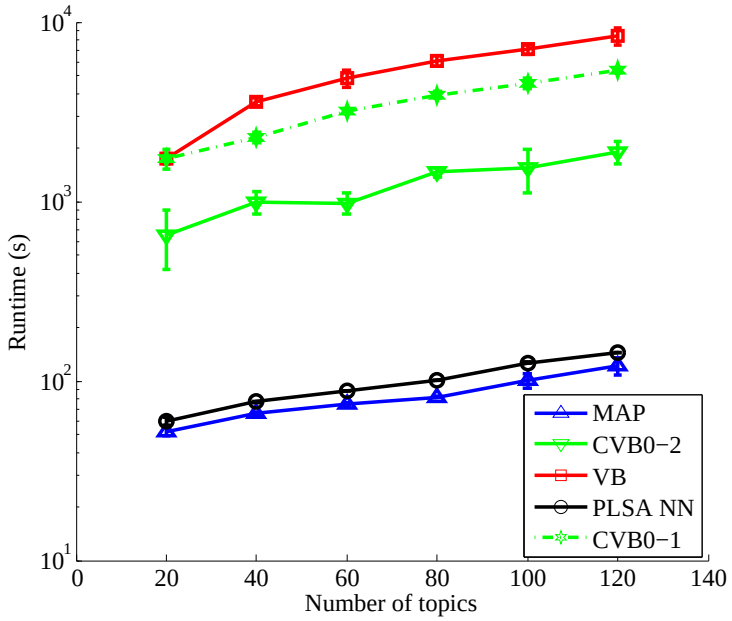


FIGURE 2.6: *LabelMe dataset:* Comparison of run time of MAP, CVB0-1, CVB0-2, variational Bayes (VB) and PLSA-NN.

As mentioned earlier, MAP can be parallelized while the collapsed inference algorithms such as CVB and CGS are not inherently parallelizable. The computational complexity of PLSA and MAP seem to be comparable in these dataset. However, for classification using PLSA, we need to compute the nearest neighbor for the test data, which can be computationally intensive for large scale applications.

2.7. Conclusion

We presented MAP, CGS and CVB0 inference algorithms for the supervised LDA model. We introduced a computationally efficient classification algorithm for

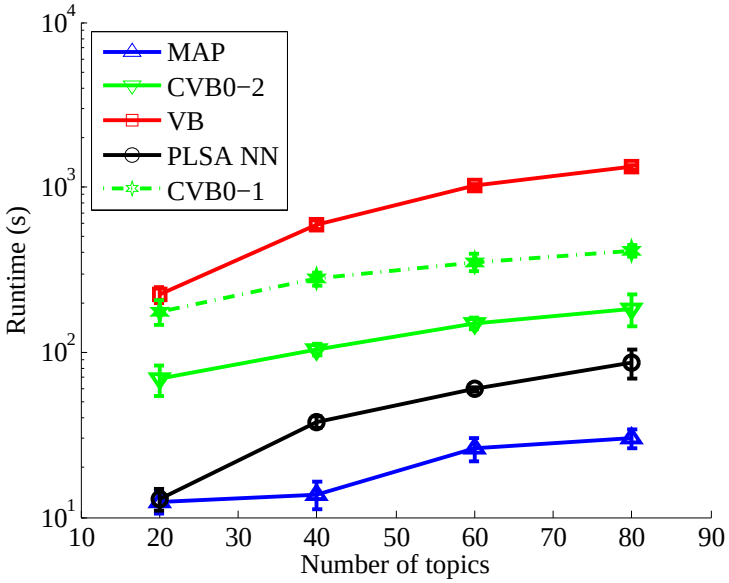


FIGURE 2.7: *MSRC-v2 dataset:* Comparison of run time of MAP, CVB0-1, CVB0-2, VB and PLSA-NN.

MAP and CVB0 that is scalable for datasets involving large number of classes. Additionally, this classification algorithm allows us to use supervised LDA as a supervised dimensionality reduction tool. We provided an empirical comparison of the classification accuracy and runtime of MAP, CVB0 to VB. The results indicate that, with proper hyperparameter tuning, CVB0 and VB can yield similar classification performance, while MAP yields a slightly worse performance. However, MAP is computationally very efficient and can provide speed-ups of over an order of magnitude compared to VB and CVB0. Additionally, we evaluated the classification performance achieved by MAP and CVB0 when topic representations obtained using LDA, as well as supervised LDA, are used as feature vector input for an SVM. The results indicate that topic representations obtained using unsupervised LDA lead to slightly higher classification accuracy than supervised LDA, and CVB0 outperforms

MAP in both supervised as well as the unsupervised case. Based on our results, we advocate CVB0 parameter estimation with the CVB0-2 classifier for the supervised LDA model, since it provides a good tradeoff between classification accuracy and run time. Future work will explore the extension of our inference algorithms to more complex topic models that can handle annotations and multiple labels.

3. PROBABILISTIC MODELS TAILORED FOR BIRD SPECIES IDENTIFICATION

3.1. Motivation

Even though different machine learning algorithms have been applied for bird species identification, there has been little work on the development of probabilistic models specific to bird vocalization. Probabilistic models enable Bayesian inference and help in identifying the interesting characteristics of data [48]. Probabilistic models have been successfully applied in other domains, for example, document clustering [49], document classification, computer vision, speech processing.

Bird vocalization is analogous to document classification in that the distribution of frame level features depends on the particular species. Hence, we can build probabilistic models for bird vocalization in a similar manner to those developed for document classification. One possibility is to treat each sound recording as a species-dependent distribution of frame level features. Another possibility is to treat each sound recording as a species-dependent distribution over syllables and treat each syllable as a distribution over frame level features. To present a weak analogy to the document classification terminology, frame level features can be viewed as words, syllables can be viewed as topic for the entire paragraph (unlike the topic assignment for every word Chapter 2. and species can be viewed as the document class. However, there are a few characteristics of bird recordings that are not usually applicable to documents. For instance, the frame level features are usually continuous in nature, the number of syllables in a specific interval of time is species-dependent, the duration of the syllable (i.e., the number of frames within

the particular syllable) is also species dependent. These differences mandate the development of probabilistic models specific to bird vocalization.

In previous work, we presented syllable-level probabilistic models [1] and Briggs et al. presented interval-level probabilistic models (interval here implies an interval of recording) [2]. The goal of this paper is to present a theoretical framework that unifies syllable-level and interval-level modeling. The main contributions of this work are:

- We present the Independent Block model and show how both syllable-level modeling and interval-level modeling are special cases of the block-level modeling approach.

- We consider two models of treating the frames within a block, namely, Independent Frame Independent Block model (IFIB) model and Markov Chain Frame Independent Block (MCFIB) model.

- We derive the Bayes risk minimizing classifier (MAP classifier) for each model and show that it can be approximated by a nearest-neighbor classifier with appropriate distance criterion, if we use a non-parametric density estimation procedure.

- For the IFIB model, we derive closed-form expressions for the distance measure for cases where the distribution of frame level features belongs to the exponential family. For the MCFIB model, we derive closed form expressions for the distance measure when the distribution of frame level features is multivariate Gaussian.

- We experimentally evaluate the accuracy of the proposed classifiers using

cross-validation on a data set consisting of 265 thirty-second recordings of six species of birds, from the Cornell Macaulay library. We compare our results to SVM. Results indicate that the proposed approaches outperform the SVM-based approach.

3.2. Problem statement

Our objective is to identify bird species based on audio recordings. We have a collection of recordings of bird sounds, each of which is labeled with a particular species. The recordings differ in their duration, so they are split into equal-length intervals. The task is to learn an acoustic model for each species based on these training set intervals so that we can correctly classify a test interval. The training dataset can be viewed as a collection of interval-label pairs, i.e., $\mathcal{S} = \{(\mathcal{D}^{(1)}, y^{(1)}), (\mathcal{D}^{(2)}, y^{(2)}), \ldots, (\mathcal{D}^{(K^t)}, y^{(K^t)})\}$, where $\mathcal{D}^{(m)}$ corresponds to the data representation of the m^{th} interval, $y^{(m)}$ denotes the label of the m^{th} interval and K^t denotes the total number of training set intervals.

We next discuss the data representation of the interval. An interval of sound can be viewed as a collection of syllables. It is common practice to divide syllables further into frames, where each frame corresponds to the sound in a very short span of time. The frames can then be represented by features such as: power spectral density, mean frequency, spectral bandwidth, short time energy, zero crossing rate, Mel frequency cepstral coefficients (MFCCs) and energy. More formally, a syllable $\mathbf{x}(i)$ consisting of n_i frames can be viewed as a sequence of observations, i.e., syllable $\mathbf{x}(i) = [x_1(i), x_2(i), \ldots, x_{n_i}(i)]$ where observation $x_j(i)$ corresponds to the feature vector representation of the j^{th} frame in the i^{th} syllable. The duration of the syllables

$$\overbrace{\underbrace{\underbrace{[x_1(1)^{(1)},\ldots,x_{n_1}(1)^{(1)}]}_{\text{Syllable 1}}\ \underbrace{\{\mathbf{x}(1)^{(1)}}_{\text{Syllable 1}}\ ,n_1^{(1)}\},\ldots,\{\mathbf{x}(N)^{(1)},n_N^{(1)}\},N^{(1)},y^{(1)}}^{\underbrace{(D^{(1)},y^{(1)})}_{\text{Interval 1}}\quad ,(D^{(2)},y^{(2)}),\ldots}}$$

FIGURE 3.1: Data representation for Syllable level modeling

$$\overbrace{[x_1^{(1)},x_2^{(1)},\ldots,x_n^{(1)}],n^{(1)},y^{(1)}}^{\underbrace{(D^{(1)},y^{(1)})}_{\text{Interval 1}}\ ,(D^{(2)},y^{(2)}),\ \ldots}$$

FIGURE 3.2: Data representation for Interval level modeling

is characteristic of a particular species, hence the number of frames within a syllable n_i is included as part of the model. The number of syllables within an interval N is also species-dependent and hence included as part of the model. Mathematically, the data in an interval of sound may be represented as $\mathcal{D} = [\mathbf{x}(1), n_1, \ldots, \mathbf{x}(N), n_N, N]$.

Fig. 3.1 explains the data hierarchy for syllable level modeling. As an alternative to syllable level modeling, one might want to model the frame level features directly at the interval-level. Fig. 3.2 explains the data hierarchy for interval level modeling. Notice that the interval level model does not distinguish between frames belonging to different syllables. The interval level models are more relevant for scenarios where segmentation can be very noisy or for applications where notion of syllables is less ambiguous. Here, we develop generative probability models for the frame level features (at both interval level and syllable level) produced by each species (from the labeled training examples) and extend these probabilistic models to build Bayes-optimal classifiers for bird species identification.

TABLE 3.1: List of symbols

Variable	Description
y	Class label (bird species)
N	Number of blocks present in a fixed interval
θ_i	Block parametrization vector of i^{th} block
n_i	Number of frames in i^{th} block (length)
$\mathbf{x}(i)$	i^{th} block features, $[x_1(i)x_2(i)\cdots x_{n_i}(i)]$
$x_j(i)$	Feature representation of the j^{th} frame (i.e., j^{th} observation) in i^{th} block
$N_y^t(l)$	Number of blocks present in the l^{th} traininginterval from class y
N_y^t	Total number of training blocks from class y, $\sum_l N_y^t(l)$
K_y^t	Number of training set intervals from class y
K^t	Total number of training set intervals
$\mathcal{D}$	Data in an interval $[\mathbf{x}(1), n_1, \ldots, \mathbf{x}(N), n_N, N]$
n	Total number of frames in an interval, $\sum_i n_i$

3.3. Probability models

In this section, we demonstrate how the interval-level and syllable-level models can be unified in a general framework. First, we define a block to be an unit of the duration of sound that lies between the duration of a frame and the duration of the interval. The size of the block is directly proportional to the number of frames within that block and inversely proportional to the number of blocks within an

interval of sound. The blocks may themselves correspond to syllables or intervals, allowing us to capture the temporal structure at different levels. A block containing n frames may be represented as $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, where x_j denotes the j^{th} frame level feature within the block. There are many possible ways to characterize a block in terms of its constituent frame level features. Here, we assume that the frame level features of the block $\mathbf{x}$ are drawn from a parametric distribution characterized by θ and use θ as the feature vector representation for the block. Next, we introduce the independent block model.

3.3.1 Independent Block model

Fig. 3.3(a) contains the independent block model. To generate an interval of sound, we follow the procedure shown in Algorithm 2.

Algorithm 2 Generative process for a single interval

Draw the class label $y \sim P(y)$

Draw the number of blocks $N \sim P(N|y)$

for $i = 1$ to N **do**

Draw the block parametrization vector $\theta_i \sim p(\theta|y)$, and the number of frames within the block $n_i \sim P(n|y)$

Draw $\mathbf{x}(i) \sim p(\mathbf{x}(i)|\theta_i, n_i)$

end for

We would like to highlight the following details about the independent block model.

- The block parametrization parameter θ_i is drawn in an i.i.d. fashion from $p(\theta|y)$.

- The independent block model does not make any assumptions about how the frame level features are drawn within the block. Here, we consider two cases where the frame level features are drawn in i.i.d. fashion or Markov chain fashion.

- Even though $p(\mathbf{x}(i)|\theta_i, n_i)$ is assumed to be a parametric probability distribution, no such restriction is imposed on $p(\theta|y)$. Here, we estimate $p(\theta|y)$ using a non-parametric density estimation procedure. This helps us model even multi-modal distributions for $p(\theta|y)$.

- One might want to choose the parametric density $p(\mathbf{x}(i)|\theta_i, n_i)$ depending on the length of the block. For short blocks, we might use unimodal distributions, whereas for long blocks, one might want to use a multi modal distribution.

3.3.2 The Interval-IID model

The Interval-IID model follows the graphical representation in Fig. 3.3(b). The model suggests that to generate an interval, we first determine its class label y based on the class prior $P(y)$. Given y, we then generate an interval-specific parameterization θ based on $p(\theta|y)$, which parameterizes the the frame feature distribution $p(x|\theta)$ of that interval. Given θ, we then generate n independent and identically distributed (i.i.d.) frame feature vectors x_j based on $p(x|\theta)$ (thus the name Interval-IID, i.e., frames are i.i.d. within an interval). Note that the Interval-IID model does not distinguish between frame level features belonging to different syllables. The Interval-IID model is a special case of the independent block model, where there is just one block ($N = 1$) and all the frame level features within the block are generated in an i.i.d. fashion. The supervised LDA model described in Chapter 2. can be viewed as a special case where the blocks correspond to the

frames. However, in that case, we first choose a document-specific distribution of topics and then choose a particular distribution before choosing each frame level feature.

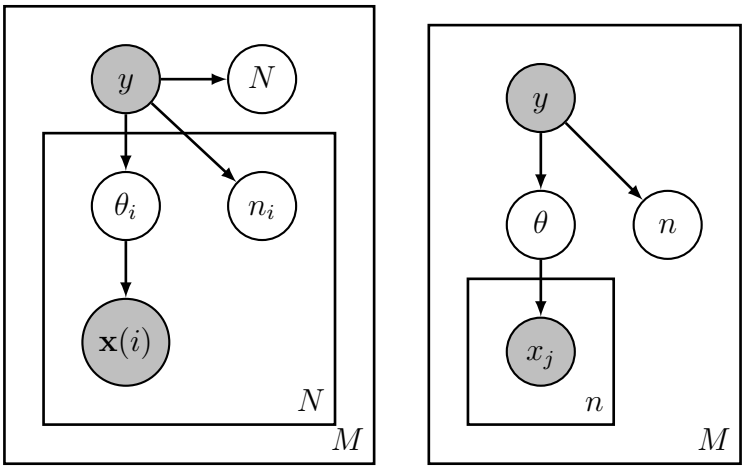
3.3.3 Independent syllable model

The independent syllable model is a special case of the independent block model where each block corresponds to a syllable. The syllables present in an interval are independent and identically distributed (i.i.d.) i.e., we select both a syllable parametrization vector θ_i (for the frame-level features) and length n_i for each syllable independent of other syllables. Note that the independent syllable model does not specify how the frame level features within a syllable are drawn. In Fig. 3.4, we present two special cases, namely,

- **Independent Frame Independent Syllable (IFIS)** model, where the frame level features are assumed to be drawn in an i.i.d. fashion conditioned on θ_i,
- **Markov Chain Frame Independent Syllable (MCFIS)** model, where the frame level features are assumed to be drawn in a Markov chain fashion conditioned on θ_i.

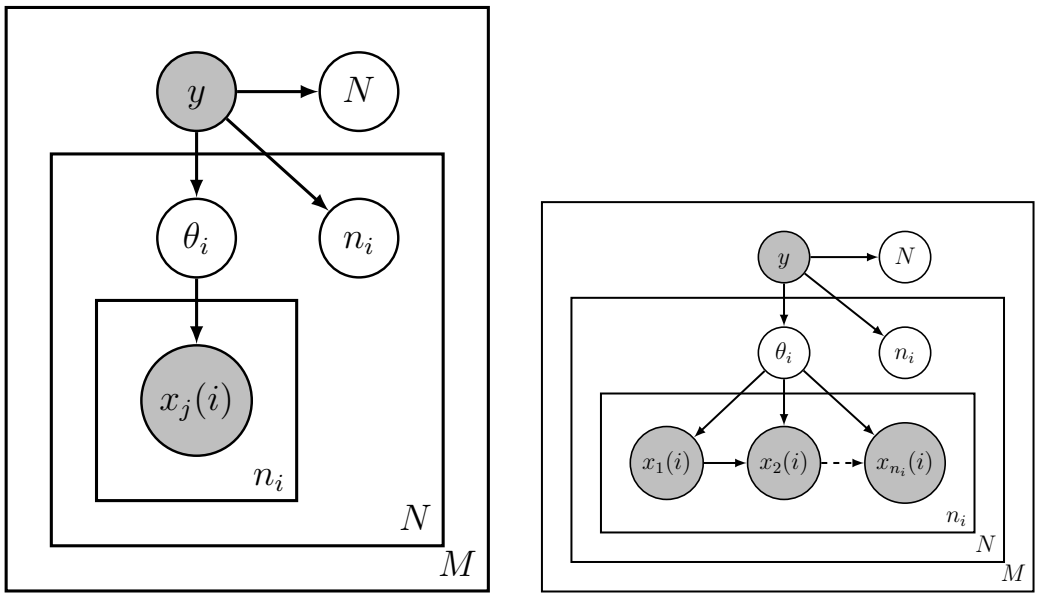
3.3.4 Taxonomy of the models

In Fig. 3.5, we present the taxonomy of the various models in a slightly different fashion from the order discussed above. We divide the models based on whether the frame level features are drawn in an i.i.d. or Markov chain fashion within the block, resulting in the independent frame independent block (IFIB) model and Markov chain frame independent block (MCFIB) model. Within the IFIB model, if the



(a) (b)

FIGURE 3.3: Graphical models of (a) the Independent Block model and (b) the Interval-IID model



(a) (b)

FIGURE 3.4: Graphical models of (a) the IFIS model and (b) the MCFIS model

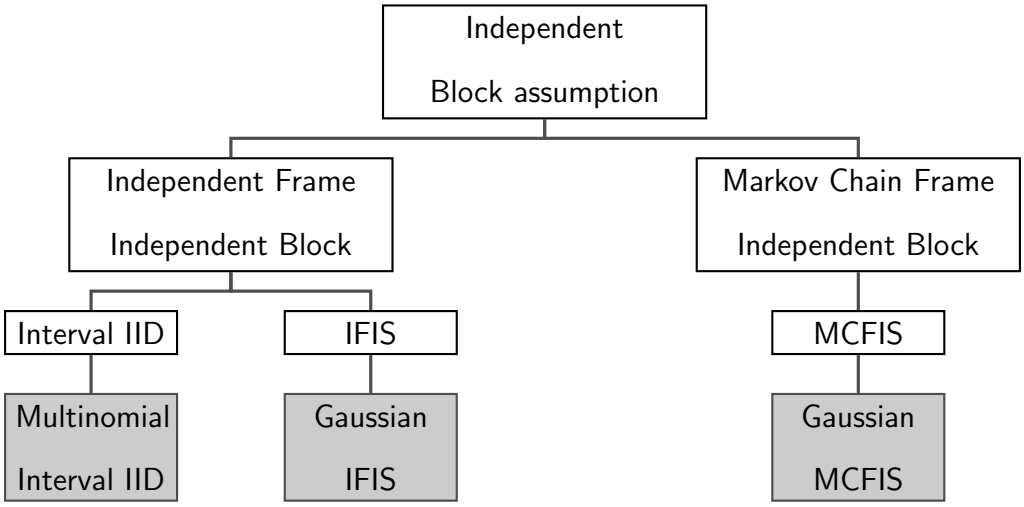


FIGURE 3.5: Taxonomy of the various models. The models considered in this work have been highlighted.

block corresponds to the interval, we obtain the Interval-IID model; if the block corresponds to syllable, we obtain the IFIS model. Within the MCFIB model, if the block corresponds to the syllable, we obtain the MCFIS model. Here, we use a multinomial distribution (multi-modal) for modeling frame-level features at the interval level and multivariate Gaussian distribution (uni-modal) for modeling frame-level features at the syllable level.

3.4. Independent Block model

Recall that the data representation for an interval is $\mathcal{D} = [\mathbf{x}(1), n_1, \ldots, \mathbf{x}(N), n_N, N]$. Based on the graphical model of the independent block model shown in Fig. 3.3(a),

the likelihood that an interval $\mathcal{D}$ was generated by class y, is given by

$$p(\mathcal{D}|y) \;=\; P(N|y)\prod_{i=1}^{N} p(\mathbf{x}(i), n_i|y) \tag{3.1}$$

The block parametrization vector θ_i is a hidden parameter and cannot be observed directly. The likelihood of a block $\mathbf{x}(i)$ can be found by marginalizing over the hidden variable θ_i i.e.,

$$p(\mathbf{x}(i), n_i|y) = E_{\theta_i|y}\Big[p(\mathbf{x}(i)|\theta_i)\Big]P(n_i|y), \tag{3.2}$$

where $E_{\theta_i|y}[\cdot]$ represents marginalization over θ_i based on the class-conditional distribution $p(\theta_i|y)$.

3.4.1 Independent Frame Independent Block (IFIB) model

As shown in Appendix E, the logarithm of Eq. (3.1) can be expressed as

$$\log p(\mathcal{D}|y) = C(\mathbf{X}) + \log P(N|y) + \sum_{i=1}^{N} \log \int_{\theta_i} e^{-n_i \hat{D}_{kl}(\hat{\theta}_i \| \theta_i)} p(\theta_i, n_i|y) d\mu(\theta_i) \tag{3.3}$$

where,

$$\hat{\theta}_i \;=\; \arg\max_{\theta_i \in \Theta} \log p(\mathbf{x}(i)|\theta_i) \tag{3.4}$$

$$\hat{D}_{kl}(\hat{\theta}_i \| \theta_i) \;=\; \frac{1}{n_i} \log \frac{p(\mathbf{x}(i)|\hat{\theta}_i)}{p(\mathbf{x}(i)|\theta_i)} \tag{3.5}$$

$$C(\mathbf{X}) \;=\; \sum_{i=1}^{N} \log p(\mathbf{x}(i)|\hat{\theta}_i) \tag{3.6}$$

Note that

- $\hat{\theta}_i$ is the maximum likelihood (ML) estimate of θ_i in the likelihood $p(\mathbf{x}(i)|\theta_i)$.
- $\hat{D}_{kl}(\hat{\theta}_i \| \theta_i)$ is a sample-estimate of the Kullback Leibler divergence (KL divergence) between the distributions $p(\cdot|\hat{\theta}_i)$ and $p(\cdot|\theta_i)$. By definition of $\hat{\theta}_i$ in Eq. (3.4), $\hat{D}_{kl}(\hat{\theta}_i \| \theta_i) \geq 0 \;\forall\; \theta_i \in \Theta$.

- $\log p(\mathbf{x}(i)|\hat{\theta}_i)$ is independent of θ_i and hence treated as a constant w.r.t. θ_i.

In the IFIB model, each block is assumed to be an i.i.d. sequence of observations, i.e., $p(\mathbf{x}(i)|\theta_i) = \prod_{j=1}^{n_i} p(x_j(i)|\theta_i)$. Hence, for the IFIB model, $\hat{D}_{kl}(\hat{\theta}_i\|\theta_i)$ in Eq. (3.5) can be expressed as

$$\hat{D}_{kl}(\hat{\theta}_i\|\theta_i) \quad = \quad \frac{1}{n_i}\sum_{j=1}^{n_i} \log \frac{p(x_j(i)|\hat{\theta}_i)}{p(x_j(i)|\theta_i)}. \tag{3.7}$$

The IFIB model is general in that it does not specify the precise form of the frame probability model $p(\cdot|\theta)$. As mentioned earlier, $\hat{D}_{kl}(\hat{\theta}_i\|\theta_i)$ can be interpreted as a sample based estimate of the true KL divergence between the distributions $p(x|\theta_i)$ and $p(x|\hat{\theta}_i)$. However, if $p(x|\theta_i)$ belongs to the exponential family of distributions, $\hat{D}_{kl}(\hat{\theta}_i\|\theta_i)$ can be shown to be equal to the true KL divergence between the distributions $p(x|\theta_i)$ and $p(x|\hat{\theta}_i)$ (See Appendix F for proof). For the IFIB model, when $p(\cdot|\theta_i)$ belongs to the exponential family, Eq. (3.3) becomes

$$\log p(\mathcal{D}|y) = C(\mathbf{X}) + \log P(N|y) + \sum_{i=1}^{N} \log \int_{\theta_i} e^{-n_i D_{kl}(\hat{\theta}_i\|\theta_i)} p(\theta_i, n_i|y) d\mu(\theta_i) \tag{3.8}$$

3.4.2 Geometric interpretation of ML

Note that if the test syllable is very similar to the training syllable, i.e., $\hat{\theta}_i \approx \theta_i$, the value of $\hat{D}_{kl}(\hat{\theta}_i\|\theta_i)$ in Eq. (3.7) approaches zero, thus maximizing the log-likelihood in Eq. (3.3).

3.4.3 Special cases of IFIB model

IFIS model

The IFIS model is a special case of the IFIB model where each block corresponds to a syllable. Next, we present a special case of the IFIS model.

Gaussian IFIS: Here, we assume that the frame probability model $p(x|\theta)$ follows a multivariate Gaussian distribution, i.e.,

$$p(x|\theta) = \frac{1}{\sqrt{\det 2\pi C}} e^{-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)}, \tag{3.9}$$

where $\theta = (\mu, C)$, i.e., the syllable parametrization vector is specified by the mean and covariance of the multivariate Gaussian distribution. $p(\cdot|\hat{\theta})$ is characterized as $p(\cdot|\hat{\mu}, \hat{C})$, where $\hat{\mu}$ and $\hat{C}$ are ML estimates of μ and C respectively. For syllable $\mathbf{x}(i) = [x_1(i), x_2(i), \ldots, x_{n_i}(i)]$, the ML estimates are given by

$$\begin{aligned} \hat{\mu} &= \frac{1}{n_i} \sum_{j=1}^{n_i} x_j(i) \\ \hat{C} &= \frac{1}{n_i} \sum_{j=1}^{n_i} (x_j(i) - \hat{\mu})(x_j(i) - \hat{\mu})^T \end{aligned}$$

If the frame level features are d-dimensional, then the number of parameters in θ is $d + d(d-1)/2 = d(d+1)/2$. If we restrict the covariance matrix to be diagonal, the number of parameters is $d + d = 2d$. This restriction is especially helpful for cases where $n_i \ll d^2$.

The multivariate Gaussian distribution belongs to the exponential family. Hence,

$$\hat{D}_{kl}\big(p(\cdot|\hat{\mu}, \hat{C}) \| p(\cdot|\mu, C)\big) = \frac{1}{2}\Big(\log \frac{\det C}{\det \hat{C}} + \mathrm{tr}(C^{-1}\hat{C} - I) + (\hat{\mu} - \mu)^T C^{-1}(\hat{\mu} - \mu)\Big),$$

where the RHS of Eq. (3.10) equals the true KL divergence between two Gaussian distributions $\mathcal{N}(\hat{\mu}, \hat{C})$ and $\mathcal{N}(\mu, C)$ [50].

Interval-IID model

The Interval-IID model is a special case of the IFIB model where $N = 1$, i.e., the entire interval is treated as a single block. For the IFIS model, the frame probability model $p(x|\theta)$ is at syllable level. However, for the Interval IID model, the frame

probability model is at the interval level. In general, the choice of unimodal frame probability model is not justified at the interval level. Hence, one might want to use a multi-modal distribution. Here, we present a case where $p(x|\theta)$ is assumed to follow a multinomial distribution.

Multinomial Interval IID: Let the frame level features be discretized into L non-intersecting bins defined by the sets A_l. Assume that frame-level feature x_j falls into one of the histogram bins $\{A_1, \ldots, A_L\}$ with probability $\{\theta_1, \ldots, \theta_L\}$, respectively. The vector $\theta = [\theta_1, \ldots, \theta_L]^T$ parametrizes a multinomial probability mass function (or a histogram), i.e., $\sum_l \theta_l = 1$ and $\theta_l \geq 0$. Hence, we have

$$p(x|\theta) = \prod_{l=1}^{L} \theta_l^{I(x \in A_l)}, \tag{3.10}$$

where $I(\cdot)$ is the indicator function which takes the value one if its argument is true and zero otherwise. Let $\hat{\theta}$ denote the ML estimate of the multinomial parameter θ. For an interval $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, $\hat{\theta}$ is given by

$$\hat{\theta}_l \;\; = \;\; \frac{1}{n} \sum_{j=1}^{n} I(x_j \in A_l)$$

If the frame level features are d-dimensional and each dimension is divided into B bins, then the number of parameters in θ is d^B. For higher dimensional frame level features, one might construct an L-dimensional codebook. We refer to [2, 6] for further details about our codebook construction procedure. Since multinomial distribution belongs to the exponential family, we have

$$\hat{D}_{kl}(\hat{\theta} \| \theta) = \sum_{l=1}^{L} \hat{\theta}_l \log \frac{\hat{\theta}_l}{\theta_l}, \tag{3.11}$$

where the RHS of Eq. (3.11) equals the true KL divergence between a multinomial distribution parameterized by $\hat{\theta}$ and another parameterized by θ.

3.4.4 Markov Chain Frame Independent Block (MCFIB) model

In the IFIB model, we considered each block to be an i.i.d. sequence of observations. Doing so, we ignored any temporal structure within the block. For instance, the i.i.d. assumption does not capture the gradient increase or decrease in mean frequency between successive frames within a syllable. A simple method to incorporate temporal structure would be to model each block as a Markov chain of observations i.e., $p(x_j(i)|x_{j-1}(i), x_{j-2}(i), \ldots, x_1(i), \theta_i) = p(x_j(i)|x_{j-1}(i), \theta_i)$. Assuming that the first observation is generated according to a probability distribution $p(\cdot|\theta)$ and denoting the conditional distribution by $p(x_j(i)|x_{j-1}(i), \theta_i)$, the likelihood of the i^{th} block can be written as product of the likelihood of the first frame and the conditional likelihood of the remaining frame-level features, i.e.,

$$p(\mathbf{x}(i)|\theta_i) = p(x_1(i)|\theta_i) \prod_{j=2}^{n_i} p(x_j(i)|x_{j-1}(i), \theta_i). \tag{3.12}$$

For tractability, we assume that $p(x_1(i)|\theta_i)$ follows an uniform distribution and is independent of y. Hence, it is irrelevant for classification.

For notational convenience, we make a slight modification to the log likelihood expression in Eq. (3.3) for the MCFIB case.

$$\log p(\mathcal{D}|y) = C(\mathbf{X}) + \log P(N|y) + \sum_{i=1}^{N} \log \int_{\theta_i} e^{-(n_i-1)\hat{D}_{kl}(\hat{\theta}_i \| \theta_i)} p(\theta_i, n_i|y) d\mu(\theta_i) \tag{3.13}$$

where $C(\mathbf{X})$, $\hat{\theta}_i$ and $\hat{D}_{kl}(\hat{\theta}_i \| \theta_i)$ are defined as

$$\begin{aligned}
C(\mathbf{X}) &= \sum_{i=1}^{N} \sum_{j=2}^{n_i} \log p(x_j(i)|x_{j-1}(i), \hat{\theta}_i) + \sum_{i=1}^{N} \log p(x_1(i))), \\
\hat{\theta}_i &= \arg\max_{\theta_i \in \Theta} \sum_{j=2}^{n_i} \log p(x_j(i)|x_{j-1}(i), \theta_i), \\
\hat{D}_{kl}(\hat{\theta}_i \| \theta_i) &= \frac{1}{n_i - 1} \sum_{j=2}^{n_i} \log \frac{p(x_j(i)|x_{j-1}(i), \hat{\theta}_i)}{p(x_j(i)|x_{j-1}(i), \theta_i)}.
\end{aligned} \tag{3.14}$$

In the MCFIB case, $\hat{\theta}_i$ for a block is obtained by maximizing the conditional likelihood $p(x_2(i), x_3(i), \ldots, x_{n_i}(i)|x_1(i), \theta)$. In the MCFIB case, $p(x_j(i)|x_{j-1}(i), \hat{\theta}_i)$ and $p(x_j(i)|x_{j-1}(i), \theta_i)$ are different distributions for each value of j. Hence the interpretation of $\hat{D}_{kl}$ as well as the definition of the KL divergence here is not as straightforward as the IFIB case.

MCFIS model

The MCFIS model is a special case of the MCFIB model where each block corresponds to a syllable. Next, we present a special case of the MCFIS model.

Gaussian MCFIS: Here, we consider the frame probability model in Eq. (3.12) to be Gaussian, i.e., $\theta_i = (\tilde{\mu}, \tilde{C})$,

$$\begin{bmatrix} x_j(i) \\ x_{j+1}(i) \end{bmatrix} \sim \mathcal{N}(\tilde{\mu}, \tilde{C}),\ \tilde{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix},\ \text{and } \tilde{C} = \begin{bmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{bmatrix}.$$

Note that in this case, the syllable parametrization vector θ includes the cross-covariance between consecutive frame level features C_{12}. For a multivariate Gaussian distribution, the conditional distribution $p(x_j(i)|x_{j-1}(i), \theta)$ is also Gaussian

$$p(x_j(i)|x_{j-1}(i), \theta_i) = \frac{1}{\sqrt{\det 2\pi C_c}} e^{-\frac{1}{2}(x_j(i)-\mu_{j|j-1})^T C_c^{-1}(x_j(i)-\mu_{j|j-1})},$$

where $\mu_{j|j-1} = \mu_2 + C_{12}^T C_{11}^{-1}(x_{j-1}(i) - \mu_1)$ and $C_c = C_{22} - C_{12}^T C_{11}^{-1} C_{12}$. If we assume the distribution of frame level features to be stationary within a syllable, we have $\mu_1 = \mu_2 = \mu$, $C_{22} = C_{11}$, and the model is characterized by $\theta = (\mu, C_{11}, C_{12})$. Here, we select $\hat{\theta}$ by maximizing the conditional likelihood as $\hat{\theta} = (\hat{\mu},\ \hat{C}_{11},\ \hat{C}_{12})$. For a test syllable $\mathbf{x}(i) = [x_1(i),\ x_2(i),\ \ldots,\ x_{n_i}(i)]$, the conditional ML solutions for $\hat{\mu}$, $\hat{C}_{11}$,

and $\hat{C}_{12}$ are given by

$$\begin{aligned}\hat{\mu} &= (I-\hat{M})^{-1}(\hat{\mu}_2 - \hat{M}\hat{\mu}_1)\\ \hat{C}_{11} &= \sum_{k=0}^{\infty}\hat{M}^k\hat{C}_c(\hat{M}^k)^T\\ \hat{C}_{12} &= \hat{C}_{11}\hat{M}^T\end{aligned}$$

where

$$\begin{aligned}\hat{M} &= \hat{C}_{12}^T\hat{C}_{11}^{-1},\\ \hat{\mu}_1 &= \frac{1}{n_i-1}\sum_{j=1}^{n_i-1}x_j(i), \quad \hat{\mu}_2 = \frac{1}{n_i-1}\sum_{j=1}^{n_i-1}x_{j+1}(i),\\ \hat{C}_{11} &= \frac{1}{n_i-1}\sum_{j=1}^{n_i-1}(x_j(i)-\mu_1)(x_j(i)-\mu_1)^T,\\ \hat{C}_{12} &= \frac{1}{n_i-1}\sum_{j=1}^{n_i-1}(x_j(i)-\mu_1)(x_{j+1}(i)-\mu_2)^T,\\ \hat{C}_c &= \hat{C}_{22} - \hat{C}_{12}^T\hat{C}_{11}^{-1}\hat{C}_{12},\\ \hat{C}_{22} &= \frac{1}{n_i-1}\sum_{j=1}^{n_i-1}(x_{j+1}(i)-\mu_2)(x_{j+1}(i)-\mu_2)^T.\end{aligned}$$

During the training phase, the model parameters μ, C_{11}, C_{12} can be estimated in a similar fashion in terms of $\mu_1, \mu_2, C_{11}, C_{12}, C_{22}$. Substituting these ML estimates in Eq. (3.14), $\hat{D}_{kl}$ for MCFIS model can be written in the following form

$$\begin{aligned}\hat{D}_{kl}(\hat{\theta}\|\theta) &= \frac{1}{2}\Big(\log\frac{\det C_c}{\det\hat{C}_c} + \mathrm{tr}(C_c^{-1}\hat{C}_c - I) + \mathrm{tr}\Big[C_c^{-1}(M-\hat{M})\hat{C}_{11}(M-\hat{M})^T\Big]\\ &\quad +(\Delta\hat{\mu}_2 - M\Delta\hat{\mu}_1)^T C_c^{-1}(\Delta\hat{\mu}_2 - M\Delta\hat{\mu}_1)\Big), \qquad (3.15)\end{aligned}$$

where $\Delta\hat{\mu}_2 = \hat{\mu}_2 - \mu_2$ and $\Delta\hat{\mu}_1 = \hat{\mu}_1 - \mu_1$. We would like to point out that $\hat{D}_{kl}(\hat{\theta}\|\theta)$ has been expressed in terms of $\mu_1, \mu_2, C_{11}, C_{12}, C_{22}$ just for convenience; inherently, the Gaussian MCFIS model involves only the parameters μ, C_{11}, C_{12}. Note that if

the test syllable is very similar to the training syllable, i.e., $\hat{\theta} \approx \theta$, the value of $\hat{D}_{kl}$ in Eq. (3.15) approaches zero, thus maximizing the log-likelihood in Eq. (3.13).

3.5. Classification and Training

We consider the Bayes risk minimizer of the probability of error for classification. Hence, our classifier is the maximum-a-posteriori (MAP) rule [48]:

$$\hat{y} = \arg\max_{y} \; p(\mathcal{D}|y)P(y), \tag{3.16}$$

which is equivalent to the maximization of the posterior $p(y|\mathcal{D})$. The log version of the MAP rule is given by

$$\hat{y} = \arg\max_{y} \; \log p(\mathcal{D}|y) + \log P(y). \tag{3.17}$$

We proceed with the evaluation of the MAP criterion for the independent block model. The MAP criterion for Interval-IID, IFIS and MCFIS models can be derived in a similar fashion. To obtain the MAP criterion for the independent block model, Eq. (3.3) is substituted into Eq. (3.17), yielding

$$\max_{y} \; \log P(y) + \log P(N|y) + \sum_{i=1}^{N} \log\Big(\int_{\theta_i} e^{-n_i \hat{D}_{kl}(\hat{\theta}_i \| \theta_i)} p(\theta_i, n_i|y) d\mu(\theta_i) \Big)$$

Typically, the models $p(\theta, n|y)$, $P(N|y)$, $P(y)$ in Eq. (3.18) are not available. We propose to estimate them from training samples in a non-parametric fashion. To estimate $p(\theta, n|y)$, we follow the kernel density estimation approach. Since only a small number of samples are available for estimating $p(\theta, n|y)$ for a given n (or potentially zero), we employ smoothing via the kernel $q(n|n(k, y))$ in our estimator:

$$\hat{p}(\theta, n|y) = \frac{1}{N_y^t} \sum_{k=1}^{N_y^t} q(n|n(k, y)) \delta(\theta - \theta(k, y)), \tag{3.18}$$

where N_y^t denotes number of training blocks from class y and $\theta(k,y)$, $n(k,y)$ respectively denote the block parametrization vector and length of the k^{th} training block from class y. The estimator $\hat{p}(\theta, n|y)$ is essentially a weighted average of the parametrization vectors of all the training blocks from class y where the weight $q(n|n(k,y))$ accounts for the block length similarity. Next, we estimate the class prior probability via the following ratio of counts

$$\hat{P}(y) = \frac{K_y^t}{K^t}, \tag{3.19}$$

where K_y^t denotes the number of training set intervals from class y and K^t denotes the total number of training set intervals. Finally, we estimate the class conditional probability for the number of blocks within an interval using the kernel density estimator

$$\hat{P}(N|y) = \frac{1}{K_y^t} \sum_{j=1}^{K_y^t} q_k(N|N(j,y)), \tag{3.20}$$

where $q_k(\cdot|\cdot)$ is the kernel and $N(j,y)$ denotes the number of blocks in the j^{th} training interval from class y. As shown in Appendix G, substituting these estimated models $\hat{p}(\theta, n|y)$, $\hat{P}(N|y)$, $\hat{P}(y)$ into Eq. (3.18) yields us the following MAP criterion

$$\begin{aligned} \min_y \quad & -\log \hat{P}(y) - \log \hat{P}(N|y) + N \log \tfrac{N_y^t}{N^t} + \textstyle\sum_{i=1}^{N} n_i d((\hat{\theta}_i, n_i) \| (\theta^{(1,i,y)}, n^{(1,i,y)})) \\ & - \textstyle\sum_{i=1}^{N} \log\Big(1 + \sum_{k=2}^{N_y^t} e^{-n_i \partial d((\hat{\theta}_i, n_i) \| (\theta^{(k,i,y)}, n^{(k,i,y)}))}\Big) \end{aligned} \tag{3.21}$$

where $d((\theta_1, n_1)\|(\theta_2, n_2))$ measures a divergence between the block parametrization vector and length of one block to that of another, and is given by

$$d((\theta_1, n_1)\|(\theta_2, n_2)) = \hat{D}_{kl}(\theta_1\|\theta_2) + d_q(n_1, n_2), \tag{3.22}$$

where $d_q(n_1, n_2)$ is a non-negative divergence for comparing block lengths and is

given by

$$d_q(n_1, n_2) = \frac{1}{n_1} \log \frac{q(n_1|n_1)}{q(n_1|n_2)}. \tag{3.23}$$

Also, $\partial d((\hat{\theta}_i, n_i) \| (\theta^{(k,i,y)}, n^{(k,i,y)}))$ in Eq. (3.21) is given by

$$\partial d((\hat{\theta}_i, n_i) \| (\theta^{(k,i,y)}, n^{(k,i,y)})) = d((\hat{\theta}_i, n_i) \| (\theta^{(k,i,y)}, n^{(k,i,y)})) - d((\hat{\theta}_i, n_i) \| (\theta^{(1,i,y)}, n^{(1,i,y)})).$$

Note the use of order statistics notation ($\theta^{(1,i,y)}$, $n^{(1,i,y)}$) to denote the nearest neighbor for the i^{th} test syllable amongst all the training examples from class y. Consider the MAP criterion in Eq. (3.21) as a sum of five terms. The first term accounts for the fact that the number of intervals from different training classes might not be equal. If all classes have equal number of training intervals, the first term becomes a constant and therefore is irrelevant to the classification. The second term accounts for the fact that the number of blocks within a fixed interval is species-dependent. The last three terms correspond to the likelihood of the observations. The last term accounts for the contribution due to training blocks other than the nearest neighbor from class y. For large n_i, the last term becomes negligible. If we consider the contribution of the nearest neighbor alone, the MAP classifier can be approximated as

$$\min_y \; - \log \hat{P}(y) - \log \hat{P}(N|y) + N \log \frac{N_y^t}{N^t} + \sum_{i=1}^{N} n_i d((\hat{\theta}_i, n_i) \| (\theta^{(1,i,y)}, n^{(1,i,y)}))$$

Note that, due to the nearest-neighbor nature of the classifier in Eq. (3.24), the training process involves only the computation of ML parameter estimates for each block in the training set. One might think of the block parametrization vector θ as feature vector and apply conventional Euclidean distance measures such as ℓ_2-norm. However, as we will show in Section 3.6., the KL divergence has connections

to Fisher information metric, which is a natural metric for comparing probability distributions. Also, the classifier in Eq. (3.24) automatically weights the distance contributions of individual blocks.

3.5.1 IFIS model

The MAP classifier for the IFIS model has the form of Eq. (3.24). In the Gaussian IFIS case, $\hat{D}_{kl}$ in Eq. (3.22) is replaced by $\hat{D}_{kl}$ from Eq. (3.10).

3.5.2 MCFIS model

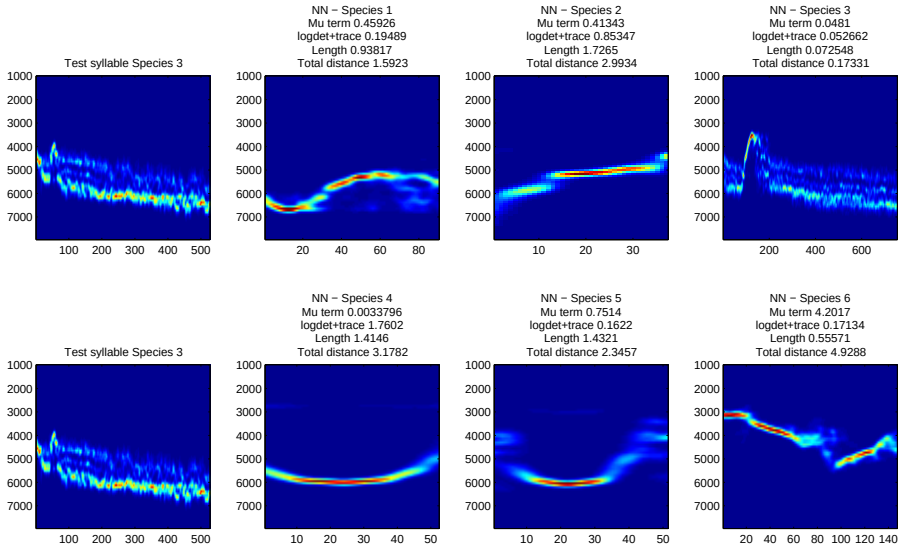
Starting with the log-likelihood as defined in Eq. (3.13), the MAP classification rule for MCFIS can be derived in a similar fashion to the independent block model. The only difference is that, for the MCFIS model, $\hat{D}_{kl}$ in Eq. (3.22) is given by Eq. (3.14) instead of Eq. (3.7). For Gaussian MCFIS, $\hat{D}_{kl}$ is given by Eq. (3.15).

Fig. 3.6 illustrates the interpretation of $\hat{D}_{kl}$ for IFIS and MCFIS models. The left most column is a test syllable from species 3. The other six elements in each figure indicate the nearest syllable from the six species based on the appropriate $\hat{D}_{kl}$. In the IFIS case, the nearest neighbors lie approximately in the same frequency range and have similar bandwidth as the original syllable, but not similar temporal structure. In the MCFIS case, the nearest neighbors tend to have a similar temporal structure as the original syllable.

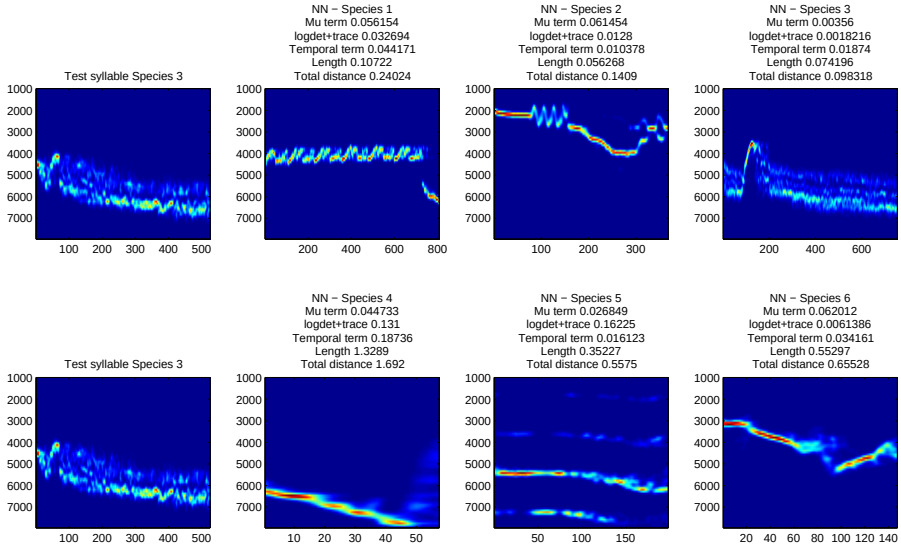
3.5.3 Interval-IID model

For Interval IID model, $N = 1$. Since $\hat{P}(N|y) = \frac{N_y^t}{N^t}$, Eq. (3.24) can be further simplified as

$$\hat{y} = \arg\min_{y} \; -\log \hat{P}(y) + d((\hat{\theta}, n) \| (\theta^{(1,y)}, n^{(1,y)})) \tag{3.24}$$



(a)



(b)

FIGURE 3.6: Interpreting $\hat{D}_{kl}$: The left most column is a test syllable from species 3. The other six elements in each figure indicate the nearest syllable from the six species based on the appropriate $\hat{D}_{kl}$. (a) IFIS (b) MCFIS

where $d((\hat{\theta}, n) \| (\theta^{(1,y)}, n^{(1,y)}))$ is defined as in Eq. (3.22). For the Multinomial-Interval IID model, $\hat{D}_{kl}$ in Eq. (3.22) is given by Eq. (3.11). The training involves just the creation of codebook and ML estimation of $\hat{\theta}$ for each training interval.

We would like to point out that when $p(x|\theta)$ is given by (3.10) and $p(\theta|y)$ is the Dirichlet distribution, then the Interval-IID model becomes the Dirichlet-Multinomial model, which is also referred to as Polya distribution [43] or the Dirichlet compound multinomial (DCM) model [49]. This model is often used as a topic model in text document classification. One criticism concerning the choice of Dirichlet prior is the limited capability of representing multi modal priors [51]. Our experience with bird sounds suggests that the probability model $p(\theta|y)$ is indeed multi modal.The non-parametric density estimation procedure employed here allows us to accommodate multi modal priors.

3.6. Nearest Neighbors on Statistical Manifolds

In this section, we explore the connection between MAP classification and the KL divergence based nearest neighbor rule. Denote a model by $p(\cdot|\theta)$ where $\theta \in \Theta$. A collection of probability models i.e., the parameter space Θ can be regarded as a manifold. The collection of models given by

$$\mathcal{M} = \left\{ p(\cdot|\theta) \mid \theta \in \Theta \in \mathbb{R}^d \right\}, \tag{3.25}$$

is a d-dimensional statistical manifold if there exists a one-to-one smooth mapping between θ to $p(\cdot|\theta)$. In the geometric approach to statistical models [52], one can measure the geodesic distance between two probability models by using the Fisher

information metric (FIM) as the Riemannian metric

$$D_{\texttt{FIM}}(p(\cdot|\theta), p(\cdot|\hat{\theta})) = \min_{\substack{\theta(\cdot), \\ \theta(0)=\theta, \\ \theta(l)=\hat{\theta}}} \int_0^l \sqrt{\dot{\theta}(t)^T \mathcal{I}(\theta(t)) \dot{\theta}(t)} dt, \tag{3.26}$$

where $\mathcal{I}(\theta)$ is the Fisher information matrix given by

$$\mathcal{I}_{ij}(\theta) = E\Big[\frac{d \log p(x|\theta)}{d\theta_i} \frac{d \log p(x|\theta)}{d\theta_j}\Big]. \tag{3.27}$$

The FIM is considered a natural metric for statistical manifolds as it reflects the capability to discriminate between probability models from their samples.

Consider a geodesic nearest neighbor rule using FIM $D_{\texttt{FIM}}(p(\cdot|\theta), p(\cdot|\hat{\theta}))$ defined in (3.26). As the precise form of the manifold is unavailable, an exact computation of the geodesic distance $D_{\texttt{FIM}}(\theta, \hat{\theta})$ is impossible. Since the nearest neighbor approach prompts us to calculate short geodesic distances, local approximations of $D_{\texttt{FIM}}(\theta, \hat{\theta})$ can be used instead. For two close probability models $\theta \to \hat{\theta}$, $D_{kl}(\hat{\theta}\|\theta) \approx D_{kl}(\theta\|\hat{\theta})$, and it is known [52] that $\sqrt{2D_{kl}(\theta\|\hat{\theta})} \to D_{\texttt{FIM}}(\theta, \hat{\theta})$. The KL divergence provides a computable approximation to the FIM manifold geodesic distance. Note that other approximations for the FIM are available (e.g., certain Ali-Silvey divergences).

3.7. Experimental Results

In this section, we describe the experimental setup used to measure the classification error rates obtained by the proposed classifiers. We first describe the implementation details of our experimental setup and then discuss the results.

3.7.1 Kernel Smoothing

We used a Poisson probability mass function (PMF) to perform kernel density estimation in both Eq. (3.20) and Eq. (3.18). In Appendix H, we present the expression for $d_q(n_1, n_2)$ when we employ a Poisson smoothing function.

3.7.2 List of Classifiers

There are many combinations of probability models and parametric assumptions. The probability models that we consider here are

- Gaussian IFIS
- Gaussian MCFIS
- Multinomial Interval-IID

For each of the above classifiers, we report results with the three possible frame level features, namely (f_c, BW), MFCCs and Spectrum PDF. For Gaussian IFIS and Gaussian MCFIS models, we restrict the covariance matrix to be diagonal for MFCCs and Spectrum PDF. For Multinomial Interval IID model, we use a codebook based approach for MFCCs and Spectrum PDF. For syllable level models, we compare the performance to a Support Vector Machine (SVM) applied at syllable level.

3.7.3 SVM setup

Support vector machines [53] (SVMs) are a family of algorithms for supervised classification that find a linear decision boundary by maximizing the margin between two classes. In cases where linear classification is insufficient, the kernel trick is applied to non-linearly project features into a higher dimensional space where

linear separability is possible. We used the Matlab interface to the LIBSVM [54] package in our experiments. Following Fagerlund [7], and the recommendations of Hsu, Chang and Lin [55], we use a radial basis function kernel, and optimize the SVM parameter C and the kernel parameter γ, by grid search. We evaluate the SVM at all combinations of C and γ in $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and report the best accuracy achieved with any set of parameters. To handle multiple classes (in our case, species), LIBSVM use the one-against-one voting scheme [56]. For Syllable-SVM, we used SVM to classify each of the individual syllables and performed a majority vote for classifying the interval based on these individual SVM decisions. Six features: 2 mean vectors, 3 unique entries from covariance matrix and syllable length were used and the features were normalized to lie in the range -1 to 1.

3.7.4 Cross Validation

To measure the accuracy of the proposed classifiers, we use them to predict the species in each of 265 thirty-second intervals of sound. Each classifier is trained using all of the intervals that do not come from the same recording as the interval being classified (the data set consists of longer recordings that are split into intervals). We use this setup so the classifier must identify species without already having example recordings of the individual bird being classified. Fagerlund [7] used a similar 'individual independent' setup for cross-validation.

Classifiers that use a codebook to construct feature histograms depend on a randomized clustering algorithm. To account for the randomness, we ran five trials for the Multinomial Interval IID model with different random seeds, and report average accuracy, $\pm$ average deviation [6].

Classifier	f_c, BW	**MFCCs**	**Spectrum PDF**
Gaussian IFIS	88.30	78.86	72.83
Gaussian MCFIS	86.03	81.88	25.28
Syllable-SVM	65.26	67.55	67.2
Multinomial Interval IID	87.32 ± 1.52	88.68 ± 0.96	85.81 ± 0.34

TABLE 3.2: The accuracy of each classifier in predicting bird species based on 265 thirty-second intervals of sound.

3.7.5 Comparison of classifiers with SVM

The results of our experiments are summarized in Table 3.2. The table contains the accuracy (correct classification rate) in % for the various models. We observe that the syllable-level models significantly outperform the Syllable-SVM in classification accuracy. The MCFIS model outperforms the IFIS model while using MFCCs, but performs worse for other frame level features. For lower dimensional features, the syllable level IFIS model outperforms the Multinomial Interval IID model. However, for higher dimensional features, the Multinomial Interval IID model significantly outperforms the syllable level models. We believe that the restricting the Gaussian covariances to be diagonal significantly affects the performance of the models. Based on our experiments, we recommend syllable level models when frame level features are low-dimensional and interval level models for high-dimensional frame level features. It is interesting to compare the performance of the Multinomial Interval IID model with the results in Table. 2.3 since these models operate on the same data. We observe that, with the right inference procedure, the

Supervised LDA model slightly outperforms the Multinomial Interval IID model.

3.8. Conclusion

We presented models tailored to bird sound recordings that can capture temporal structure, and additional information such as the duration and frequency of vocalizations. The models are very general in that they can be used to model the temporal structure at the syllable level as well as the interval level. We used a non-parametric density estimation procedure and showed that the MAP classifier can be interpreted as a nearest neighbor classifier. We presented experimental results that indicate the ability of our models to outperform SVM-based approaches. The nearest neighbor approach allows us to visualize similar patterns from existing database of recordings and can be used for information retrieval purposes as well.

4. CONCLUSION

4.1. Summary

In this thesis, we presented probabilistic models and inference techniques for classification of bioacoustic data. We demonstrated that probabilistic models can be used model various sources of information in an elegant way. By using principled inference techniques, probabilistic models provide a refreshing alternative to generic classifiers that require efficient feature vector transformations and extensive tuning procedures. We developed a wide variety of inference techniques for our models. The insights from our experiments enable us to choose the most suitable inference procedure depending on the requirements of the application (eg., classification performance, training time, test time). The ability of probabilistic models to model different types of data and trade-off performance measures with a suitable inference technique, make them powerful tools for efficiently modeling the rich variety of bioacoustic data.

4.2. Contributions

The contributions in this thesis are listed below.

1. Syllable level probabilistic models for bird species identification [1].
2. Efficient inference techniques for parameter estimation and classification in supervised Latent Dirichlet Allocation model [36].

3. General class of probabilistic models for bird species identification that can be applied either at the syllable level or the interval level [6].

4.3. Publications

The following publications were published as part of M.Sc work.

4.3.1 Journal publications

1. B. Lakshminarayanan, F. Briggs, R. Raich, and X.Z. Fern, "Probabilistic Models for Audio Classification of Bird Species," *In preparation*

4.3.2 Conference publications

1. B. Lakshminarayanan, R. Raich, and X. Fern, "A Syllable-Level Probabilistic Framework for Bird Species Identification," in *International Conference on Machine Learning and Applications*. IEEE, 2009, pp. 53–59

2. B. Lakshminarayanan and R. Raich, "Non-negative matrix factorization for parameter estimation in hidden markov models," in *IEEE Workshop on Machine Learning for Signal Processing (MLSP),*, 2010, pp. 89–94

3. B. Lakshminarayanan and R. Raich, "Inference in Supervised Latent Dirichlet Allocation," *Submitted, Under review*

4.4. Future work

One obvious extension would be extend our models to classify MIML data. LDA-style models have been developed for MIML problems [32, 58]. It would be in-

teresting to compare the computational complexity and classification performance of the different inference techniques for these MIML models. Most of the topic models use generative criterion such as maximum marginal likelihood for parameter estimation. It would be interesting to develop efficient approximate inference techniques for discriminative training criteria. Most prior work in MIML evaluates classification performance at the bag-level. Here, we are also interested in classifying the syllables and recovering the species-specific vocabulary from MIML training data. Developing and evaluating models that achieve good instance-level classification by learning from bag-level labels is a very interesting future direction.

BIBLIOGRAPHY

1. B. Lakshminarayanan, R. Raich, and X. Fern, "A Syllable-Level Probabilistic Framework for Bird Species Identification," in *International Conference on Machine Learning and Applications*. IEEE, 2009, pp. 53–59.

2. 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*. IEEE, 2009, pp. 51–60.

3. Panu Somervuo, Aki Härmä, and Seppo Fagerlund, "Parametric representations of bird sounds for automatic species recognition," in *IEEE Transactinos on Audio, Speed, and Language Processing*. 2006, vol. 14, IEEE Press.

4. C. Catchpole and P. J. B. Slater, *Bird song: biological themes and variations*, Cambridge University Press, 1995.

5. Seppo Fagerlund, *Automatic Recognition of Bird Species by their Sounds*, Ph.D. thesis, Helsinki University of Technology, 2004.

6. B. Lakshminarayanan, F. Briggs, R. Raich, and X.Z. Fern, "Probabilistic Models for Audio Classification of Bird Species," *In preparation.*

7. S. Fagerlund, "Bird species recognition using support vector machines," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 64–64, 2007.

8. Chang-Hsing Lee, Yeuan-Kuen Lee, and Ren-Zhuang Huang, "Automatic recognition of bird songs using cepstral coefficients," *Journal of Information Technology and Applications*, vol. 1, no. 1, pp. 17 – 23, 2006.

9. M. Xu, L.Y. Duan, J. Cai, L.T. Chia, C. Xu, and Q. Tian, "HMM-based audio keyword generation," *Advances in Multimedia Information Processing-PCM 2004*, pp. 566–574, 2005.

10. Z.H. Zhou and M.L. Zhang, "Multi-instance multi-label learning with application to scene classification," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1609, 2007.

11. 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*, vol. 100, no. 2, pp. 1209–1219, 1996.

12. A. L. McIlraith and H. C. Card, "Birdsong recognition using backpropagation and multivariatestatistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2740–2748, 1997.

13. A. Selin, "Wavelets in recognition of bird sounds," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–9, 2007.

14. A. Härmä, "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*, 2003, vol. 5, pp. V–545–8 vol.5.

15. 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 *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 5, pp. 289–292.

16. Panu Somervuo and Aki Härmä, "Analyzing bird song syllables on the self-organizing map," in *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, Kitakyushu, Japan, Sept. 2003.

17. Joseph A. Kogan and Daniel Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study," *The Journal of the Acoustical Society of America*, vol. 103, no. 4, pp. 2185–2196, 1998.

18. P. Somervuo and A. Harma, "Bird song recognition based on syllable pair histograms," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP04*, 2004, vol. 5, pp. 825–828.

19. D.J. Hu and L.K. Saul, "A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles," 2009.

20. Claus Seyerlehner, Gerhard Widmer, and Peter Knees, "Frame Level Audio Similarity - A Codebook Approach," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, September 1–4 2008.

21. T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM New York, NY, USA, 1999, pp. 50–57.

22. D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

23. D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

24. T.L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. Suppl 1, pp. 5228, 2004.

25. Y.W. Teh, D. Newman, and M. Welling, "A collapsed variational bayesian inference algorithm for latent dirichlet allocation," *Advances in neural information processing systems*, vol. 19, pp. 1353, 2007.

26. A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh, "On smoothing and inference for topic models," in *Uncertainty in Artificial Intelligence*, 2009, vol. 100.

27. L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 2005, pp. 524–531.

28. C. Wang, D. Blei, and L. Fei-Fei, "Simultaneous image classification and annotation," in *Proc. CVPR*, 2009.

29. L. Du, L. Ren, B. Dunson, and L. Carin, "A Bayesian Model for Simultaneous Image Clustering, Annotation and Object Segmentation," *Advances in Neural Information Processing Systems*, vol. 22, pp. 486–494, 2009.

30. S. Lacoste-Julien, F. Sha, and M.I. Jordan, "DiscLDA: Discriminative learning for dimensionality reduction and classification," *Advances in Neural Information Processing Systems 21 (NIPS08)*, 2008.

31. J. Zhu, A. Ahmed, and E.P. Xing, "MedLDA: maximum margin supervised topic models for regression and classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM New York, NY, USA, 2009.

32. S.H. Yang, H. Zha, and B.G. Hu, "Dirichlet-Bernoulli Alignment: A Generative Model for Multi-Class Multi-Label Multi-Instance Corpora," *Advances in Neural Information Processing Systems*, 2009.

33. D.M. Blei and J. McAuliffe, "Supervised topic models," *Advances in Neural Information Processing Systems*, vol. 20, pp. 121–128, 2008.

34. G. Xu, S.H. Yang, and H. Li, "Named entity mining from click-through data using weakly supervised latent dirichlet allocation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2009, pp. 1365–1374.

35. D. Ramage, D. Hall, R. Nallapati, and C.D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2009, pp. 248–256.

36. B. Lakshminarayanan and R. Raich, "Inference in Supervised Latent Dirichlet Allocation," *Submitted, Under review*.

37. H.M. Wallach, D. Mimno, and A. McCallum, "Rethinking LDA: Why priors matter," in *Topic Models: Text and Beyond Workshop in Neural Information Processing Systems Conference*, 2009.

38. D. Mimno and A. McCallum, "Topic models conditioned on arbitrary features with dirichlet-multinomial regression," in *Proc. of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

39. D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2001.

40. E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005, pp. 601–602.

41. M.J. Beal, *Variational algorithms for approximate Bayesian inference*, 2003.

42. S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge Univ Pr, 2004.

43. T. Minka, "Estimating a Dirichlet distribution," 2003.

44. M. Girolami and A. Kabán, "On an equivalence between PLSI and LDA," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, p. 434.

45. A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," *Computer Vision–ECCV 2006*, pp. 517–530, 2006.

46. David Arthur and Sergei Vassilvitskii, "k-means++: the advantages of careful seeding," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, PA, USA, 2007, pp. 1027–1035, Society for Industrial and Applied Mathematics.

47. C.C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines," 2001.

48. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

49. C. Elkan, "Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution," in *Proceedings of the 23rd international conference on Machine learning*. ACM New York, NY, USA, 2006, pp. 289–296.

50. J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proc. ICASSP*, 2007, vol. 4, pp. 317–320.

51. K. Yu, S. Yu, and V. Tresp, "Dirichlet enhanced latent semantic analysis," in *Conference in Artificial Intelligence and Statistics*, 2005.

52. R.E. Kass and P.W. Vos, *Geometrical foundations of asymptotic inference*, Wiley-Interscience, 1997.

53. Corinna Cortes and Vladimir Vapnik, "Support vector networks," in *Machine Learning*, 1995, pp. 273–297.

54. Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001.

55. C.W. Hsu, C.C. Chang, C.J. Lin, et al., "A practical guide to support vector classification," 2003.

56. C.W. Hsu and C.J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

57. B. Lakshminarayanan and R. Raich, "Non-negative matrix factorization for parameter estimation in hidden markov models," in *IEEE Workshop on Machine Learning for Signal Processing (MLSP),*, 2010, pp. 89–94.

58. S.H. Yang, J. Bian, and H. Zha, "Hybrid Generative/Discriminative Learning for Automatic Image Annotation," .

APPENDIX

A Segmentation algorithm

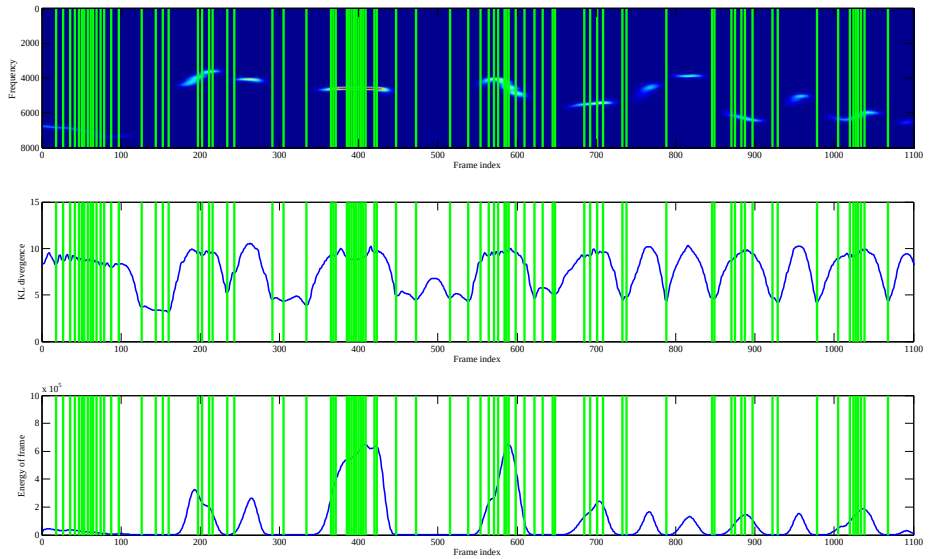
We first compute the spectrogram of the audio signal and then compute the power spectral density (PSD) of each frame and normalize it to obtain the normalized PSD. Next, we compute the Kullback Leibler (KL) divergence between the normalized power spectral density (PSD) of each frame and the uniform distribution. We use the locations of local minima of the KL divergence to determine boundaries between elements. The regions within the boundaries are treated as elements. For each element, we compute the average energy as well as the average KL divergence of the constituent frames. Elements for which both the average energy as well as average KL divergence are above a threshold are treated as 'potential syllables'. We set the threshold for average energy (average KL divergence) to be 1% of the maximum value of the average energy (average KL divergence) of all elements within that interval. Finally, if two or more 'potential syllables' are adjacent, they are merged to get the list of final syllables. Fig. 0.1 and 0.2 contain the segmentation outputs for audio recordings belonging to Winter Wren and Swainson's Thrush respectively.

B Derivation of MAP objective function for Supervised LDA

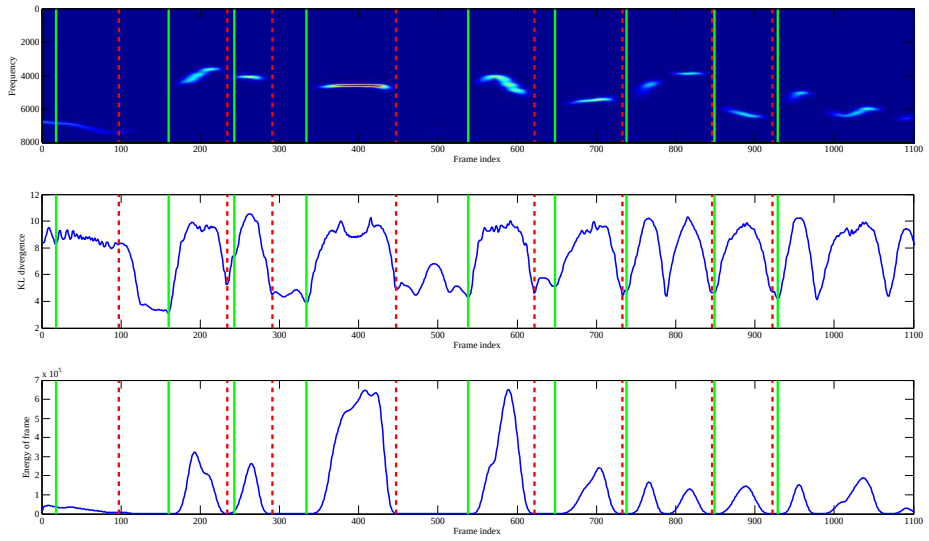
Based on the structure of the graphical model, we have

$$p(\mathbf{W}, \boldsymbol{\theta}, \mathbf{Y}, \boldsymbol{\phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\phi}|\boldsymbol{\beta}) \prod_{i=1}^{M} p(\mathbf{w}_i, \theta_i, y_i | \boldsymbol{\alpha}, \boldsymbol{\phi}). \tag{B.1}$$

Next, we derive the expressions for each term on the RHS of (B.1). Let $\mathbf{z}_i$ denote the respective topic assignments for each word in the vector $\mathbf{w}_i$. For each document,

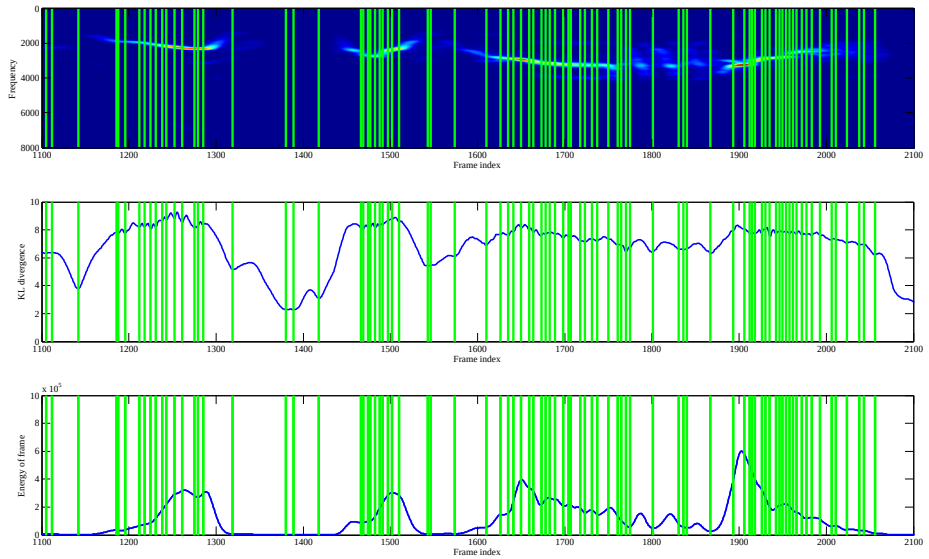


(a)

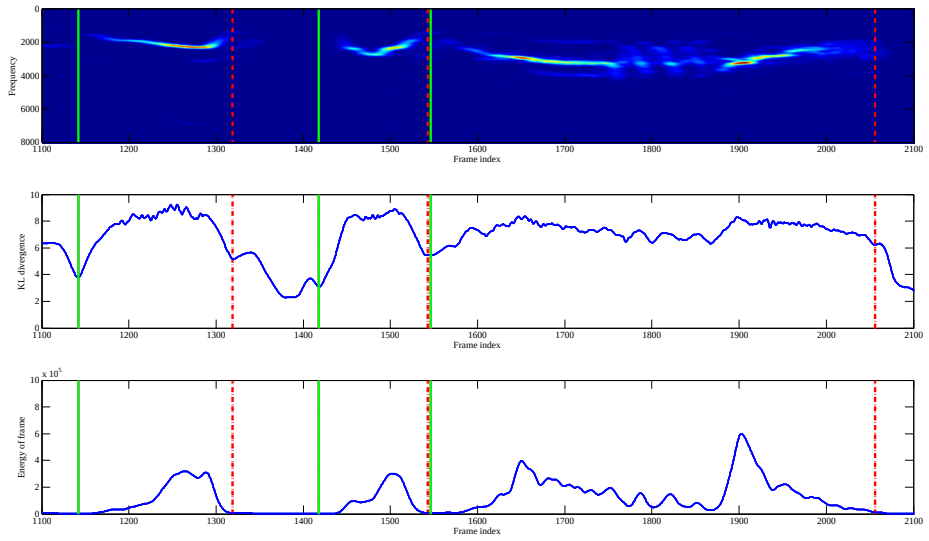


(b)

FIGURE 0.1: Segmentation of Winter Wren recording: (a) The local minima in KL divergence are marked initially (b) Each region between the initial set of markers is treated as an element. The final list of syllables are obtained using the procedure described in Section A. The start and end of the final syllables are marked in green and red respectively.



(a)



(b)

FIGURE 0.2: Segmentation of Swainson's Thrush recording: (a) The local minima in KL divergence are marked initially (b) Each region between the initial set of markers is treated as an element. The final list of syllables are obtained using the procedure described in Section A. The start and end of the final syllables are marked in green and red respectively.

we can marginalize out all possible topic assignments, i.e.,

$$p(\mathbf{w}_i, \theta_i, y_i|\boldsymbol{\alpha}, \boldsymbol{\phi}) \;=\; \Big(\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i, \boldsymbol{\phi}) P(\mathbf{z}_i|\theta_i)\Big) \cdot p(\theta_i|y_i, \boldsymbol{\alpha}) P(y_i).$$

The first term on the RHS of (B.2) can be simplified as

$$\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i, \boldsymbol{\phi}) P(\mathbf{z}_i|\theta_i) = \exp(\sum_{v=1}^{V} n_{vi} \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi}), \tag{B.2}$$

where n_{vi} denotes the number of times word v occurs in the i^{th} training document. Using the fact that $p(\theta_i|y_i, \boldsymbol{\alpha})$ and $p(\phi_{k,.}|\boldsymbol{\beta})$ are Dirichlet distributions, we obtain (2.2)

C Derivation of (B.2)

Here, we discuss the marginalization of $\mathbf{z}_i$ in the first term on the RHS of (B.2).

$$\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i, \boldsymbol{\phi}) P(\mathbf{z}_i|\theta_i) \;=\; \sum_{\mathbf{z}_i} \prod_{j=1}^{N_i} \prod_{k'=1}^{K} \Big(\theta_{ik'} \prod_{v=1}^{V} \phi_{k'v}^{\mathbb{1}[w_{ij}=v]}\Big)^{\mathbb{1}[z_{ij}=k']}.$$

Due to the independence structure, the marginalization can be done independently for each z_{ij},

$$\sum_{\mathbf{z}_i} P(\mathbf{w}_i|\mathbf{z}_i,\boldsymbol{\phi})P(\mathbf{z}_i|\theta_i) = \prod_j \sum_{z_{ij}=1}^{K} \prod_{k'=1}^{K} \Big[\theta_{ik'}\ \phi_{k',w_{ij}}\Big]^{\mathbb{1}[z_{ij}=k']}.$$

Replacing z_{ij} by k,

$$= \prod_j \sum_{k=1}^{K} \theta_{ik}\ \phi_{k,w_{ij}}$$

Representing $\sum_{k=1}^{K} \theta_{ik}\ \phi_{k,w_{ij}}$ as an inner product,

$$\begin{aligned} &= \prod_j (\boldsymbol{\phi}^\top \boldsymbol{\theta})_{w_{ij},i}, \\ &= \exp(\sum_j \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{w_{ij},i}), \\ &= \exp(\sum_j \sum_{v=1}^{V} \Big[\mathbb{1}[w_{ij}=v] \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi}\Big]), \\ &= \exp(\sum_{v=1}^{V} \Big[n_{vi} \log(\boldsymbol{\phi}^\top \boldsymbol{\theta})_{vi}\Big]), \end{aligned}$$

where $n_{vi} = \sum_{j=1}^{N_i} \mathbb{1}[w_{ij}=v]$.

D Derivation of MAP update equations in (2.3)

Here, we present the co-ordinate optimization of $\boldsymbol{\theta}$ in (2.2). $\boldsymbol{\phi}$ can be optimized in a similar fashion. For a fixed $\boldsymbol{\phi}$, the Lagrange function (including the sum-to-one constraints on $\boldsymbol{\theta}$) is given by

$$J(\boldsymbol{\theta}) = \sum_i \Big(\sum_{v=1}^{V} \Big[n_{vi} \log(\boldsymbol{\phi}^T \boldsymbol{\theta})_{vi}\Big] + \sum_{k=1}^{K} (\alpha_{y_i,k} - 1) \log \theta_{ik}\Big) - \sum_i \lambda_i (\sum_k \theta_{ik} - 1) + \text{constant},$$

where 'constant' refers to terms independent of $\boldsymbol{\theta}$. First, we derive a lower bound for $\log(\boldsymbol{\phi}^T\boldsymbol{\theta})_{vi}$ using Jensen's inequality.

$$\log(\boldsymbol{\phi}^T\boldsymbol{\theta})_{vi} = \log(\sum_{k=1}^{K}\phi_{kv}\theta_{ik}) = \log(\sum_{k=1}^{K}\gamma_k\frac{\phi_{kv}\theta_{ik}}{\gamma_k}).$$

Assume that $\gamma_k \geq 0\ \forall\ k$ and $\sum_{k=1}^{K}\gamma_k = 1$. Applying Jensen's inequality, we have

$$\log(\boldsymbol{\phi}^T\boldsymbol{\theta})_{vi} \geq \sum_{k=1}^{K}\gamma_k\log(\frac{\phi_{kv}\theta_{ik}}{\gamma_k}).$$

Defining γ_k as

$$\gamma_k = \frac{\phi_{kv}\hat{\theta}_{ik}}{\sum_{k'=1}^{K}\phi_{k'v}\hat{\theta}_{ik'}},$$

we obtain

$$\begin{aligned} J(\boldsymbol{\theta}) \ \geq\ & \sum_i\Big(\sum_{v=1}^{V}\Big[n_{vi}\sum_{k=1}^{K}\frac{\phi_{kv}\hat{\theta}_{ik}}{\sum_{k'=1}^{K}\phi_{k'v}\hat{\theta}_{ik'}}\log\theta_{ik}\Big] + \sum_{k=1}^{K}(\alpha_{y_i,k}-1)\log\theta_{ik}\Big) \\ & - \sum_i\lambda_i(\sum_k\theta_{ik}-1) + \text{constant}. \end{aligned}$$

Note that the surrogate (lower bound) is separable in $\theta'_{ik}s$ if $\boldsymbol{\phi}$ is held constant. Re-arranging the order of summations,

$$\begin{aligned} J(\boldsymbol{\theta}) \ \geq\ & \sum_i\sum_{k=1}^{K}\Big(\Big[\sum_{v=1}^{V}n_{vi}\frac{\phi_{kv}\hat{\theta}_{ik}}{\sum_{k'=1}^{K}\phi_{k'v}\hat{\theta}_{ik'}} + (\alpha_{y_i,k}-1)\Big]\log\theta_{ik}\Big) \\ & - \sum_i\lambda_i(\sum_k\theta_{ik}-1) + \text{constant}. \end{aligned}$$

Imposing the non-negativity constraints on $\boldsymbol{\theta}$, we obtain the update equations in (2.3).

E Nearest neighbor form of the log likelihood

Substituting Eq. (3.2) in Eq. (3.1) and using $p = e^{\log p}$, we obtain

$$p(\mathcal{D}|y) = P(N|y)\prod_{i=1}^{N}E_{\theta_i|y}\Big[e^{\log p(\mathbf{x}(i)|\theta_i)}\Big]P(n_i|y). \tag{E.1}$$

We can express $\log p(\mathbf{x}(i)|\theta_i)$ as follows:

$$\log p(\mathbf{x}(i)|\theta_i) = \log p(\mathbf{x}(i)|\hat{\theta}_i) - n_i \hat{D}_{kl}(\hat{\theta}_i \| \theta_i), \tag{E.2}$$

where $\hat{\theta}_i$ and $\hat{D}_{kl}(\hat{\theta}_i \| \theta_i)$ have been defined in Eq. (3.4) and (3.5) respectively. Substituting Eq. (E.2) in the logarithm of Eq. (E.1), we obtain

$$\log p(\mathcal{D}|y) = C(\mathbf{X}) + \log P(N|y) + \sum_{i=1}^{N} \Big(\log P(n_i|y) + \log E_{\theta_i|y} \Big[e^{-n_i \hat{D}_{kl}(\hat{\theta}_i \| \theta_i)} \Big] \Big),$$

where $C(\mathbf{X})$ has been defined in Eq. (3.6). Using the integral form of expectation, we obtain

$$\log p(\mathcal{D}|y) = C(\mathbf{X}) + \log P(N|y) + \sum_{i=1}^{N} \Big(\log P(n_i|y) + \log \int_{\theta_i} e^{-n_i \hat{D}_{kl}(\hat{\theta}_i \| \theta_i)} p(\theta_i|y) d\mu(\theta_i) \Big),$$

from which we obtain Eq. (3.3).

F $\hat{D}_{kl}(\hat{\theta}\|\theta)$ for Exponential family (i.i.d. case)

The exponential family is characterized as

$$p(x|\theta) = h(x) e^{\eta(\theta)^T T(x) - A(\eta(\theta))}$$

where $\eta(\theta)$ corresponds to the natural parametrization of the exponential family.

For block $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ containing n frames, if we assume that the frames are i.i.d. within the block, we have $p(\mathbf{x}|\theta) = \prod_{j=1}^{n} p(x_j|\theta)$. Let $\hat{\theta}$ denote the maximum likelihood estimator, i.e.,

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{n} \log p(x_j|\theta)$$

For the exponential family, we have the following property, [48]

$$E_{\hat{\theta}}[T(x)] = \frac{1}{n}\sum_{j=1}^{n} T(x_j). \tag{F.1}$$

The true KL divergence between two distributions belonging to the same exponential family is given by

$$\begin{aligned} D_{kl}(\hat{\theta}\|\theta) &= \int_x p(x|\hat{\theta}) \log \frac{p(x|\hat{\theta})}{p(x|\theta)} dx \\ &= E_{\hat{\theta}}\Big[\log \frac{p(x|\hat{\theta})}{p(x|\theta)}\Big] \\ &= \Big(\eta(\hat{\theta}) - \eta(\theta)\Big)^T E_{\hat{\theta}}[T(x)] - A(\eta(\hat{\theta})) + A(\eta(\theta)) \end{aligned} \tag{F.2}$$

Next, we evaluate the sample KL divergence estimate,

$$\begin{aligned} \hat{D}_{kl}(\hat{\theta}\|\theta) &= \frac{1}{n}\sum_{j=1}^{n} \log \frac{p(x_j|\hat{\theta})}{p(x_j|\theta)} \\ &= \frac{1}{n}\sum_{j=1}^{n} \Big(\eta(\hat{\theta}) - \eta(\theta)\Big)^T T(x_j) - A(\eta(\hat{\theta})) + A(\eta(\theta)) \\ &= \Big(\eta(\hat{\theta}) - \eta(\theta)\Big)^T \Big(\frac{1}{n}\sum_{j=1}^{n} T(x_j)\Big) - A(\eta(\hat{\theta})) + A(\eta(\theta)) \end{aligned} \tag{F.3}$$

From Eq. (F.1), (F.3), (F.2), we have $\hat{D}_{kl}(\hat{\theta}\|\theta) = D_{kl}(\hat{\theta}\|\theta)$ for any $p(x|\theta)$ belonging to the exponential family.

G Learning using kernel density estimates

Substituting the estimates $\hat{p}(\theta, n|y)$, $\hat{P}(N|y)$, $\hat{P}(y)$ into Eq. (3.18) and adding a constant (w.r.t y) term $N \log N^t$ for convenience, we obtain the MAP criterion in a new form

$$\max_y \;\; \log \hat{P}(y) + \log \hat{P}(N|y) - N \log \frac{N_y^t}{N^t} + \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{N_y^t} q(n_i|n(k,y)) e^{-n_i \hat{D}_{kl}(\hat{\theta}_i\|\theta(k,y))}\Big).$$

Combining terms to a single exponent

$$\max_y \ \log \hat{P}(y) + \log \hat{P}(N|y) - N \log \frac{N_y^t}{N^t} + \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{N_y^t} e^{-n_i(\hat{D}_{kl}(\hat{\theta}_i \| \theta(k,y)) + \frac{1}{n_i} \log \frac{1}{q(n_i|n(k,y))})}\Big).$$

Subtracting the constant (w.r.t. y) $\sum_{i=1}^{N} \log q(n_i|n_i)$:

$$\max_y \ \log \hat{P}(y) + \log \hat{P}(N|y) - N \log \frac{N_y^t}{N^t} + \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{N_y^t} e^{-n_i(\hat{D}_{kl}(\hat{\theta}_i \| \theta(k,y)) + \frac{1}{n_i} \log \frac{q(n_i|n_i)}{q(n_i|n(k,y))})}\Big).$$

Let $d((\theta_1, n_1)\|(\theta_2, n_2)) = \hat{D}_{kl}(\theta_1\|\theta_2) + \frac{1}{n_1} \log \frac{q(n_1|n_1)}{q(n_1|n_2)}$ and note that if $q(n|l)$ is uniquely maximized at $n = l$, then $\frac{1}{n_1} \log \frac{q(n_1|n_1)}{q(n_1|n_2)} \geq 0$ and is zero if and only if $n_1 = n_2$. Hence $d_q(n_1, n_2) = \frac{1}{n_1} \log \frac{q(n_1|n_1)}{q(n_1|n_2)}$ is a divergence between n_1 and n_2. In this case, $d((\theta_1, n_1)\|(\theta_2, n_2))$ being the sum of two divergences is also a divergence which applied to (θ_1, n_1) and (θ_2, n_2). Hence, we can further express the MAP criterion as

$$\max_y \ \log \hat{P}(y) + \log \hat{P}(N|y) - N \log \frac{N_y^t}{N^t} + \sum_{i=1}^{N} \log\Big(\sum_{k=1}^{N_y^t} e^{-n_i d((\hat{\theta}_i, n_i)\|(\theta(k,y), n(k,y)))}\Big)$$

Let $(\theta^{(k,i,y)}, n^{(k,i,y)})$ denote the ordered version of $(\theta(k,y), n(k,y))$ w.r.t. k, such that

$$d((\hat{\theta}_i, n_i)\|(\theta^{(1,i,y)}, n^{(1,i,y)})) \leq d((\hat{\theta}_i, n_i)\|(\theta^{(2,i,y)}, n^{(2,i,y)})) \leq \ldots$$
$$\leq d((\hat{\theta}_i, n_i)\|(\theta^{(N_y^t,i,y)}, n^{(N_y^t,i,y)}))$$

Using the order notations and applying a negative sign, we rewrite the MAP criterion as

$$\min_y \ -\log \hat{P}(y) - \log \hat{P}(N|y) + N \log \frac{N_y^t}{N^t} - \sum_{i=1}^{N} \log \sum_{k=1}^{N_y^t} \Big(e^{-n_i d((\hat{\theta}_i, n_i)\|(\theta^{(1,i,y)}, n^{(1,i,y)}))}\Big)$$

Expressing $d((\hat{\theta}_i, n_i)\|(\theta^{(k,i,y)}, n^{(k,i,y)})$ as

$$d((\hat{\theta}_i, n_i)\|(\theta^{(k,i,y)}, n^{(k,i,y)}) = d((\hat{\theta}_i, n_i)\|(\theta^{(1,i,y)}, n^{(1,i,y)}) + \partial d((\hat{\theta}_i, n_i)\|(\theta^{(k,i,y)}, n^{(k,i,y)}),$$

we obtain Eq. (3.21).

H Kernel smoothing function - Poisson case

Consider the Poisson kernel function given by

$$q(n|l) = \frac{l^n e^{-l}}{n!}$$

The corresponding divergence is given by

$$\begin{aligned} d_q(n,l) &= \frac{1}{n}\log\frac{q(n|n)}{q(n|l)} \\ &= \frac{1}{n}\log\frac{n^n e^{-n}}{l^n e^{-l}} \\ &= \frac{l}{n} - 1 - \log\frac{l}{n}, \end{aligned}$$

which can be verified to be non-negative as $\log(x) \leq (x-1)$.