HaMMLeT: An Infinite Hidden Markov Model with Local Transitions

by

Colin Reimer Dawson

A Dissertation Submitted to the Faculty of the

Program in Statistics

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy

In the Graduate College

The University of Arizona

2 0 1 6

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we have read the dissertation

prepared by Colin Reimer Dawson

entitled "HaMMLeT: An Infinite Hidden Markov Model with Local Transitions"

and recommend that it be accepted as fulfulling the dissertation requirement for the

Degree of Doctor of Philosophy.

_____ Date: 08/08/2016
Clayton Morrison

_____ Date: 08/08/2016
Katherine Barnes

_____ Date: 08/08/2016
Helen Zhang

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfulling the dissertation requirement.

_____ Date: 08/08/2016
Dissertation Director: Clayton Morrison

Statement by Author

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Signed: Colin Reimer Dawson

Acknowledgements

I would like to acknowledge the following people for their support, intellectual, moral, emotional, and otherwise.

- Clay Morrison, my advisor, for his tireless support, for helping me stay organized, and for challenging me to be able to explain myself.
- Kobus Barnard, for turning me on to Bayesian Modeling in the first place, way back in the aughts sometime, and for providing the resources and opportunities for me to actually learn to put ideas to code.
- The rest of my committee, Kathie Barnes and Helen Zhang, for their valuable feedback.
- Paul Cohen, for bringing me in to SISTA, providing the opportunity (and funding!) to work on so many varied challenging research projects, and giving me so much leeway in teaching for the information science program.
- Kyle Simek, Ernesto Brau, and Andrew Prehoehl, for being excellent lab mates, for putting together the summer "coding bootcamps" that were so valuable to me as someone without a lot of formal CS training, for the very stimulating reading groups, and for being there to bounce ideas off of.
- My first research student at Oberlin, Bill Huang, for writing code behind several of the figures herein.
- My parents, for being able to smile and nod when I said I was going to do *another* Ph.D., and for not asking "have you started writing your dissertation yet?" more than once a week.
- My children, Arlo and Esai, for being there with snuggles and laughter every day, and reminding me what's actually important.
- My wife, Sarah, for everything and more. There are no words.

Dedication

To Sarah, my amazing wife, for putting up with me during not one, but *two* rounds of graduate school. Next, the tenure track. What fun for her!

Table of Contents

TABLE OF CONTENTS—*Continued*

TABLE OF CONTENTS—*Continued*

List of Tables

List of Figures

List of Figures—*Continued*

List of Figures—*Continued*

List of Figures—*Continued*

Abstract

In classical mixture modeling, each data point is modeled as arising i.i.d. (typically) from a weighted sum of probability distributions. When data arises from different sources that may not give rise to the same mixture distribution, a hierarchical model can allow the source contexts (e.g., documents, sub-populations) to share components while assigning different weights across them (while perhaps coupling the weights to "borrow strength" across contexts). The Dirichlet Process (DP) Mixture Model (e.g., Rasmussen (2000)) is a Bayesian approach to mixture modeling which models the data as arising from a countably infinite number of components: the Dirichlet Process provides a prior on the mixture weights that guards against overfitting. The Hierarchical Dirichlet Process (HDP) Mixture Model (Teh et al., 2006) employs a separate DP Mixture Model for each context, but couples the weights across contexts. This coupling is critical to ensure that mixture components are reused across contexts.

An important application of HDPs is to time series models, in particular Hidden Markov Models (HMMs), where the HDP can be used as a prior on a doubly infinite transition matrix for the latent Markov chain, giving rise to the HDP-HMM (first developed, as the "Infinite HMM", by Beal et al. (2001), and subsequently shown to be a case of an HDP by Teh et al. (2006)). There, the hierarchy is over rows of the transition matrix, and the distributions across rows are coupled through a top-level Dirichlet Process.

In the first part of the dissertation, I present a formal overview of Mixture Models and Hidden Markov Models. I then turn to a discussion of Dirichlet Processes and their various representations, as well as associated schemes for tackling the problem of doing approximate inference over an infinitely flexible model with finite computational resources. I will then turn to the Hierarchical Dirichlet Process (HDP) and its application to an infinite state Hidden Markov Model, the HDP-HMM.

These models have been widely adopted in Bayesian statistics and machine learning. However, a limitation of the vanilla HDP is that it offers no mechanism to model correlations between mixture components across contexts. This is limiting in many applications, including topic modeling, where we expect certain components to occur or not occur together. In the HMM setting, we might expect certain states to exhibit similar incoming and outgoing transition probabilities; that is, for certain rows and columns of the transition matrix to be correlated. In particular, we might expect pairs of states that are "similar" in some way to transition frequently to each other. The HDP-HMM offers no mechanism to model this similarity structure.

The central contribution of the dissertation is a novel generalization of the HDP-HMM which I call the Hierarchical Dirichlet Process Hidden Markov Model With Local Transitions (HDP-HMM-LT, or HaMMLeT for short), which allows for correlations between rows and columns of the transition matrix by assigning each state a location in a latent similarity space and promoting transitions between states that are near each other. I present a Gibbs sampling scheme for inference in this model, employing auxiliary variables to simplify the relevant conditional distributions, which have a natural interpretation after re-casting the discrete time Markov chain as a continuous time Markov Jump Process where holding times are integrated out, and where some jump attempts "fail". I refer to this novel representation as the Markov Process With Failed Jumps. I test this model on several synthetic and real data sets, showing that for data where transitions between similar states are more common, the HaMMLeT model more effectively finds the latent time series structure underlying the observations.

HaMMLeT: An Infinite Hidden Markov Model with Local Transitions

Colin Reimer Dawson, Ph.D.
The University of Arizona, 2016

Director: Clayton Morrison

In classical mixture modeling, each data point is modeled as arising i.i.d. (typically) from a weighted sum of probability distributions. When data arises from different sources that may not give rise to the same mixture distribution, a hierarchical model can allow the source contexts (e.g., documents, sub-populations) to share components while assigning different weights across them (while perhaps coupling the weights to "borrow strength" across contexts). The Dirichlet Process (DP) Mixture Model (e.g., Rasmussen (2000)) is a Bayesian approach to mixture modeling which models the data as arising from a countably infinite number of components: the Dirichlet Process provides a prior on the mixture weights that guards against overfitting. The Hierarchical Dirichlet Process (HDP) Mixture Model (Teh et al., 2006) employs a separate DP Mixture Model for each context, but couples the weights across contexts. This coupling is critical to ensure that mixture components are reused across contexts.

An important application of HDPs is to time series models, in particular Hidden Markov Models (HMMs), where the HDP can be used as a prior on a doubly infinite transition matrix for the latent Markov chain, giving rise to the HDP-HMM (first developed, as the "Infinite HMM", by Beal et al. (2001), and subsequently shown to be a case of an HDP by Teh et al. (2006)). There, the hierarchy is over rows of the transition matrix, and the distributions across rows are coupled through a top-level Dirichlet Process.

In the first part of the dissertation, I present a formal overview of Mixture Models and Hidden Markov Models. I then turn to a discussion of Dirichlet Processes and

their various representations, as well as associated schemes for tackling the problem of doing approximate inference over an infinitely flexible model with finite computational resources. I will then turn to the Hierarchical Dirichlet Process (HDP) and its application to an infinite state Hidden Markov Model, the HDP-HMM.

These models have been widely adopted in Bayesian statistics and machine learning. However, a limitation of the vanilla HDP is that it offers no mechanism to model correlations between mixture components across contexts. This is limiting in many applications, including topic modeling, where we expect certain components to occur or not occur together. In the HMM setting, we might expect certain states to exhibit similar incoming and outgoing transition probabilities; that is, for certain rows and columns of the transition matrix to be correlated. In particular, we might expect pairs of states that are "similar" in some way to transition frequently to each other. The HDP-HMM offers no mechanism to model this similarity structure.

The central contribution of the dissertation is a novel generalization of the HDP-HMM which I call the Hierarchical Dirichlet Process Hidden Markov Model With Local Transitions (HDP-HMM-LT, or HaMMLeT for short), which allows for correlations between rows and columns of the transition matrix by assigning each state a location in a latent similarity space and promoting transitions between states that are near each other. I present a Gibbs sampling scheme for inference in this model, employing auxiliary variables to simplify the relevant conditional distributions, which have a natural interpretation after re-casting the discrete time Markov chain as a continuous time Markov Jump Process where holding times are integrated out, and where some jump attempts "fail". I refer to this novel representation as the Markov Process With Failed Jumps. I test this model on several synthetic and real data sets, showing that for data where transitions between similar states are more common, the HaMMLeT model more effectively finds the latent time series structure underlying the observations.

1. STATISTICAL MODELS AND MODEL SELECTION

1.1. Introduction

The statistician George Box famously said that "Essentially, all models are wrong, but some are useful." (Box et al., 1987). That is, by necessity, a statistical model is a simplification of the phenomenon that it is modeling, and indeed, increasing fidelity is not always a good thing. For one, more complex models are more difficult to understand, which can be important if the goal of modeling is to achieve some insight about the law-like behavior governing a natural phenomenon. But even if the model is being applied strictly for the purposes of making predictions, and being able to gain some verbal understanding is not a priority, the more complex a model becomes, the more sensitive it is to the idiosyncracies of the data that it is informed by.

An important question in statistics is how to go about choosing a model with enough complexity and flexibility to account for the interesting regularities in the process that produced the data, but not so much complexity that the model will not generalize; after all, the purpose of statistical modeling is almost always to explain something and/or to make predictions about the future, not simply to describe data already collected.

A number of techniques have been developed in both frequentist and Bayesian statistics to select from among a set of model forms one that balances fit and generalizability, that is, to do **model selection**. An alternative to model selection is to employ a model form whose complexity is not fixed, but instead adapts to become increasingly flexible as the data set grows larger: this is the **nonparametric** approach, which has a long history in frequentist statistics (e.g., Rosenblatt et al. (1956); Parzen (1962)), where the typical approach is to model the data without any assumptions about the specific distributional family that generated the observations.

Distribution-free estimators have been shown to have good asymptotic properties (see, e.g., Wasserman (2007) for a review of the theory), but for finite samples, usually require *ad hoc* "smoothing" parameters to be specified by hand, or chosen using cross-validation. In Bayesian statistics, "smoothing" occurs naturally through the prior, but the notion of a model with no distributional assumptions at all is not particularly natural in a statistical approach that requires a well-defined distribution to be specified over all possible data generating processes.

In recent decades, however, and especially in the last decade, a field of **Bayesian nonparametrics** has seen a great deal of development. Bayesian nonparametric models share with their non-Bayesian counterparts the property that the flexibility of the model increases as the sample size increases, but instead of being entirely distribution-free, Bayesian nonparametric models employ countably infinite dimensional families of distributions that can approximate large classes of distributions (e.g., all continuous distributions) arbitrarily well (Lindsay, 1995; McLachlan and Peel, 2004). The result is to combine the explicitness of assumptions in Bayesian models with the flexibility of nonparametric models.

The central innovation in this dissertation is a new nonparametric Bayesian model for sequence data, the Hierarchical Dirichlet Process Hidden Markov Model with Local Transitions (HDP-HMM-LT, or HaMMLeT, for short). Hidden Markov Models (HMMs) are used to model observed sequence data as "noisy" realizations of an underlying sequence of discrete states. The HaMMLeT model generalizes an existing model, the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM), by allowing a metric structure to be specified on the state space of the underlying Markov chain, providing a mechanism to build into the state transition prior the notion that transitions should be more likely between states that are "similar" in some way.

A challenge for most Bayesian modeling, not least inifinte-dimensional nonparametric Bayesian modeling, is developing efficient algorithms to compute (or approximate) desired functions of the posterior distribution, since for all but the simplest

models, explicit integration is intractable. For nonparametric models in particular, a key property of any practical inference algorithm is that it must be able to represent needed properties of the infinite-dimensional posterior using finite storage and processing. Thus, any time a new nonparametric model is proposed, a central challenge is introducing a tractable and efficient inference algorithm. I present a simple Gibbs sampling algorithm (Geman and Geman, 1984) to sample from the joint posterior of all parameters and hyperparameters of HaMMLeT as well as the latent state sequence, which is achieved by the introduction of a set of auxiliary variables with a natural interpretation under an augmentation of the stochastic process underlying HaMMLeT.

1.1.1. An overview of the dissertation

In the remainder of this chapter, I begin by discussing the general problem of model selection in the context of the bias variance tradeoff, and how Bayesian models in particular provide a natural Occam's Razor principle to balance fidelity to data on the one hand, and simplicity on the other. Then I introduce the finite dimensional versions of the models on which HaMMLeT will build, namely mixture models and their sequential counterparts, Hidden Markov Models. In Chapter 2 I review the theory of Bayesian nonparametrics, in particular Dirichlet Processes and Hierarchical Dirichlet Processes, which underlie infinite mixture models and the infinite-state Hidden Markov Model, respectively, and then define the HDP-HMM of which HaMMLeT is a generalization. In Chapter 3 I introduce the basic theory of the HaMMLeT model and an augmented stochastic process, the Markov Jump Process with Failed Transitions, which provides an interpretable set of auxiliary variables that greatly simplify posterior inference in HaMMLeT. Then in Chapters 4-5 I define some specific instantiations of HaMMLeT for a variety of kinds of data, and give experimental results comparing HaMMLeT's inferential performance to existing models. Finally, in 6, I

give concluding thoughts and some directions for future research.

1.2. Model Selection

1.2.1. The likelihood function and the conservation of explanatory power

In a parametric model family, $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$, the principle of maximum likelihood estimation can be seen as choosing the parameter vector θ that makes the data as *unsurprising* as possible. One consequence of this principle is that models that are consistent with a broad range of possible data sets must spread their predictive probability mass over a larger volume of possible data, and thus assign a smaller probability (or probability density) to any given data set. When comparing the likelihood for a more restrictive model to a more flexible model when both are consistent with the data, then, the likelihood function will tend to prefer the restrictive one.

To see a trivial example of this idea, consider the inference problem of trying to decide what "grammatical rule" produced the following set of sequences:

lay lay dee

way way dee

fay fay dee

A few candidate rules are the following:

1. Any three syllables (call this the X Y Z rule)
2. Any repeated syllable followed by another syllable (call this X X Y)
3. Any repeated syllable followed by the syllable dee (X X dee)
4. Any repeated syllable rhyming with "day", followed by the syllable dee (*ay *ay dee)

5. Three repetitions of the same syllable (XXX).

Which one seems to best describe the pattern? All but the XXX rule are consistent with all three examples, but if we attribute a probability model to each rule, by, for example, supposing that the content of each "free variable" is chosen uniformly from the set of viable candidates in the language, then we see that the *ay *ay dee rule is the maximum likelihood choice, being the most restrictive which is still consistent with the data.

In a sense, the likelihood principle exhibits the property of *conservation of explanatory power*: a model has a total probability mass that it can use to "place bets" on possible data sets. Models that allocate their probability mass over many data sets are less likely to be categorically wrong, but do not get as big a "reward" as a model that distributes its bet over a smaller number of possibilities. Like betting on roulette, the "vague" bet on red is reasonably likely to pay off, but results in a smaller profit than the narrow bet on 34, which is much less likely to result in a payout, but when it does, it is a large one. On the other hand, a model that makes precise predictions is akin to betting on 28: most conceivable outcomes result in outright rejection (zero reward), but a small number yield a big boost.

This property of rewarding the most restrictive models acts as a form of "Occam's Razor", which is the principle that states that all else being equal we should prefer simpler explanations. But although the likelihood function rewards simpler (or at least more restrictive) explanations in these toy examples of competing models with no free parameters, it breaks down when competing models have differing numbers of "moving parts", due to the problem of **overfitting**: the more flexible a model is, the more likely it can fit idiosyncracies of the data, treating them as signal when they are more appropriately considered noise. Since maximum likelihood minimizes surprise, it will tend to give high scores to models that purport to "explain" as much of the variability in the data as possible, leaving as little as possible to random chance

(noise).

1.2.2. The bias-variance tradeoff

The notion of a tradeoff between model complexity and sensitivity to idiosyncracies in the data is formalized by the **bias-variance tradeoff**.

Suppose that we have a model class, $\mathcal{F}$, and using some data, $\{X, Y\}$, we will choose a specific model $\hat{f} \in \mathcal{F}$ that can be used to make a prediction about some future observation, y_{new} given some features, x_{new}. That is, $\hat{f}(x_{new})$ yields some predicted value $\hat{y}_{new}$, which we want to be as close as possible to the true but unknown value y_{new}.

Typically we cannot make perfect predictions, and there will be some error, which we might quantify using squared distance:

$$\mathsf{Error}(x_{new}) = (\hat{f}(x_{new}) - y_{new})^2$$

For a given model fitting criterion or procedure (such as minimizing squared distance), each dataset yields a potentially different solution in the form of the fitted model $\hat{f}$, and thus a different amount of prediction error, and so we can consider the *expected* error in prediction, or the **mean squared error** (MSE)

$$MSE = \mathbb{E}[(\hat{f}(x_{new}) - y_{new})^2]$$

where the expectation is taken with respect to the true (but unknown) distribution of the data.

We can decompose the MSE as follows:

$$\begin{aligned} MSE &= \mathbb{E}[(\hat{f}(x_{new}) - \mathbb{E}[\hat{f}(x_{new})] + \mathbb{E}[\hat{f}(x_{new})] - \mathbb{E}[y \mid x_{new}] + \mathbb{E}[y \mid x_{new}] - y_{new})^2] \\ &= \mathbb{V}[\hat{f}(x_{new})] + (\mathbb{E}[\hat{f}(x_{new})] - \mathbb{E}[y \mid x_{new}])^2 + \mathbb{V}[y_{new} \mid x_{new}] \\ &= \mathbb{V}[\hat{f}(x_{new})] + \mathsf{Bias}^2(\hat{f}(x_{new})) + \sigma_\varepsilon^2(x_{new}) \end{aligned}$$

where the cross terms all cancel out. The first term in the result describes how much the prediction made by the fitted model varies depending on the specific dataset used to choose $\hat{f}$. The second is the squared **bias**, where the bias of a model is the average discrepancy between its prediction and the truth. Finally, σ_ε^2 describes the unavoidable indeterminacy of trying to make a prediction about y_{new} using only x_{new}. This last term does not depend on the choice of model class, $\mathcal{F}$, but the bias and the variance do.

In general, the more complex the model class $\mathcal{F}$, the more flexibility it has to fit the true pattern of the data, and so the greater its ability to be unbiased; however, this freedom often comes at the expense of greater variance: even if on average the model class yields a correct prediction, for a given dataset its prediction may be farther away from that correct "average behavior". This phenomenon, of excessive sensitivity to idiosyncracies of the particular dataset, is known as **overfitting**. On the other hand, we can trivially produce a model with zero variance by using a constant prediction regardless of the data; but this model is presumably going to have greater bias, since unless y_{new} is not related at all to the input x_{new}, there will be values of the latter for which the corresponding y values are not centered around our constant prediction (and even if the mean of y is constant, we are unlikely to be able to intuit this without using any data).

1.2.3. Example: polynomial regression

We can see the tradeoff between bias and variance (and the consequences of overfitting) in the setting of fitting a polynomial curve of order d to a data with a one dimensional target variable and a single input dimension. In Fig. 1.1, we see the resulting fit to a sample of 12 data points drawn from a polynomial model with $d = 5$ and independent Normally distributed residuals for a model class with d equal to 1, 2, and 10. The linear model has low variance, but high bias: it underfits the data. The

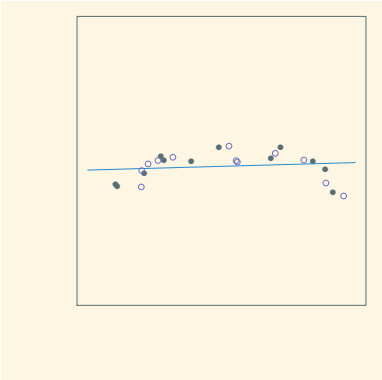
tenth order model overfits the data, having high variance and low bias: it achieves a perfect fit on the sample used to select parameters, but we would get dramatically different predictions if the sample included the unfilled points rather than the filled points. In fact, as we see in Fig. 1.1d, the predictions made for the unfilled points by the curve fit to the filled points are poor; and the converse would be true as well. The model that achieves the lowest out-of-sample prediction error is a model of intermediate complexity; in this case, just one degree away from the true model used to generate the data.

1.2.4. Balancing fit and complexity: Bayesian Occam's razor

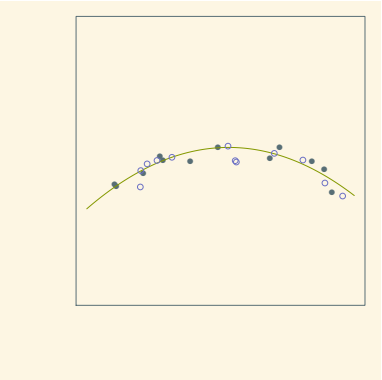
How do we go about deciding how complex a model to use in a given setting? One common approach is to use a penalized fit statistic, in which the specific model f is chosen not to maximize raw fit to the data (e.g., by maximizing the likelihood), in which case more complex models that contain simpler models as special cases will always be chosen, but instead to maximize a *penalized* objective function that balances in-sample fit with some measure of complexity. The use of a penalized objective is known as *regularization*. This complexity penalty can be viewed as a means to hold down the variance of the model class; i.e., to combat overfitting.

Some examples of penalized fit measures that serve to select a subset of predictor variables to receive nonzero coefficients in linear regression are the lasso (L_1-penalized regression), Mallow's C_p, and various "information criteria" such as AIC and BIC (Akaike, 2011; Schwarz, 1978).

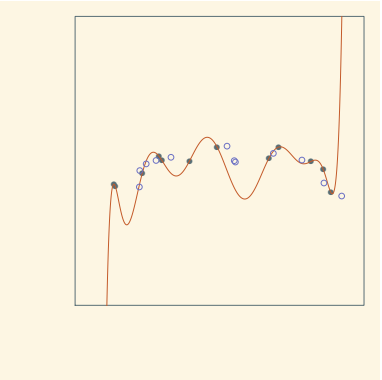
Bayesian inference provides an alternative approach. Whereas the likelihood function serves as a measure of how surprising the data is given a fully specified model, Bayesian inference provides a method for quantifying how surprising the data and the specific model are together, where the data is assumed to be selected stochastically from a set of possible datasets allowed by the model (as in maximum likelihood



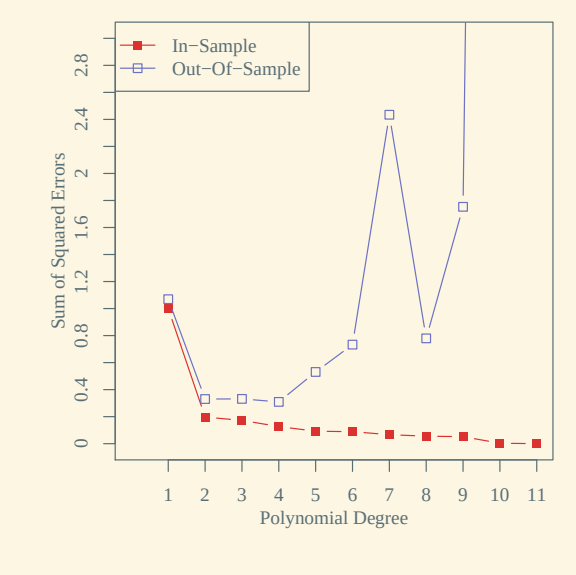
(a) Linear Fit



(b) Quadratic Fit



(c) Tenth-Order Fit



(d) Errors on Training and Test Set

Figure 1.1: Polynomials of varying degrees fit using ordinary least squares (equivalently, maximum likelihood estimation) to a set of eleven points generated from a fifth order polynomial with independent Normal residuals. 1.1a–1.1c: fits of polynomial order 1, 2 and 10. Filled circles are training data, unfilled circles are data not used during fitting. 1.1d squared prediction error in and out of the training sample for polynomial orders 1 through 10.

estimation), and additionally, the *model* is treated as though it were selected stochastically from a set of possible models. At each of these "choice points" (choosing a model from the space of models, and receiving a dataset from the set of possible datasets, if there are lots of options, any one of them becomes more surprising.

Consider the problem of choosing between two model classes, $\mathcal{M}_1$ and $\mathcal{M}_2$, where $\mathcal{M}_1$ has no free parameters, and $\mathcal{M}_2$ has a free parameter θ, and reproduces $\mathcal{M}_1$ for a particular setting of $\theta = \theta_*$. The likelihood function by itself gives us no way by itself to adjudicate between the two equivalent models, $\mathcal{M}_1$ on the one hand, and $\mathcal{M}_2$ with $\theta = \theta_*$ on the other since, $p(Y \mid \mathcal{M}_1) = p(Y \mid \mathcal{M}_2, \theta_*)$. Bayes' rule yields

$$p(\mathcal{M}_1 \mid Y) = \frac{p(\mathcal{M}_1)p(Y \mid \mathcal{M}_1)}{p(Y)} \tag{1.1}$$

$$p(\mathcal{M}_2 \mid Y) = \frac{p(\mathcal{M}_2)p(Y \mid \mathcal{M}_2)}{p(Y)} \tag{1.2}$$

where in the case of $\mathcal{M}_2$, we need to expand the likelihood to

$$p(Y \mid \mathcal{M}_2) = \int_\Theta p(Y \mid \theta, \mathcal{M}_2)p(\theta \mid \mathcal{M}_2)d\theta \tag{1.3}$$

which is a weighted average of specific likelihoods of the form $p(Y \mid \theta, \mathcal{M}_2)$ with weights given by some distribution over the parameter space. Thus, if $\mathcal{M}_1$ assigns a large likelihood to the observed data, even if $\mathcal{M}_2$ can achieve the same likelihood at θ_*, the "good value" θ_* will be diluted in the **marginal likelihood**, $p(Y \mid \mathcal{M}_2)$.

Hence, unlike maximum likelihood, Bayes theorem will tend to "reward" more restrictive *model classes* even in the case that the more flexible model contains the simpler one as a special case, providing a built in Occam's Razor. Of course, if there is some θ other than θ_* for which the likelihood is greater, then $\mathcal{M}_2$ may win out depending on whether the benefit to the likelihood outweighs the "penalty" induced by the $p(\theta \mid \mathcal{M}_2)$ term, which is what we want to happen, since the simplest model is only to be preferred if it does about as good a job accounting for the data as the more complex model.

I now turn to the example of clustering a set of observations into some number of groups, or, similarly, of fitting a mixture distribution to a dataset. I will introduce this problem in some detail, since it is closely related to the main topic of this dissertation, namely Hidden Markov Models, which I will introduce in the following section, and

treat in more detail in Chapter 2. I introduce the clustering application here since it helps to motivate the bigger picture idea of a nonparametric model.

1.3. Clustering via a Mixture of Gaussians Model

Suppose we have a model of the form

$$f(y \mid \pi, \theta) = \sum_{k=1}^{\infty} \pi_k f(y \mid \theta_k) \tag{1.4}$$

where $\theta = (\theta_1, \theta_2, \dots)$ are some parameters governing the mixture components, and $\pi = (\pi_1, \pi_2, \dots)$ is the vector of mixing weights that sum to 1.

We could take a Bayesian approach and put priors on the vector of mixture weights π and on each θ_k.

Concretely, suppose that f is Normal, so that $\theta_k = (\mu_k, \sigma_k^2)$, and first suppose that we use a prior on π_k that fixes the number of nonzero weights to a finite value, K. Then the model is of the form

$$f(y \mid \pi, \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \sigma_k^2) \tag{1.5}$$

The π_i define a categorical distribution over K categories. The categorical family has the conjugate prior

$$f(\pi \mid \xi) \propto \prod_{k=1}^{K} \pi_k^{\xi_k - 1} \tag{1.6}$$

which is a **Dirichlet distribution**.

We can also place conjugate priors on the μ_k and σ_k, such as Normal and Inverse-Gamma, in the case of Normal components.

(Note that when y is vector valued, μ becomes a mean vector and σ^2 becomes a covariance matrix. It is also possible to define conjguate priors for these.)

Putting everything together and assuming Normal components, we have the joint prior:

$$f(\pi, \mu, \sigma^{-2}) \propto \prod_{k=1}^{K} \pi_k^{\xi_k - 1} e^{-\frac{1}{2\sigma_k^2}(\mu_k - \mu_0)^2} \sigma_k^{-2a_0 - 2} e^{-\sigma_k^{-2} b_0} \tag{1.7}$$

and the joint likelihood

$$f(y \mid \pi, \mu, \sigma^2) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \sigma_k^{\frac{1}{2}} e^{-\frac{1}{2\sigma_k^2}(x_i - \mu_k)^2} \tag{1.8}$$

That sum is trouble for inference. What we can do is to represent the likelihood in two stages: instead of saying that each y_i is drawn from a weighted mixture of the K components, we can instead say that each y_i comes from precisely one component; we just do not know which one. However, we can say that y_i comes from component k with probability k. We can imagine generating data as follows:

For each $i = 1, \ldots, n$

1. Draw $z_i \sim \text{Categorical}(\pi_1, \ldots, \pi_K)$
2. Draw $y_i \sim f(y \mid \mu_{z_i}, \sigma_{z_i}^2)$

The joint likelihood for the z_i and y_i becomes

$$f(z_i, y_i \mid \pi, \mu, \sigma^2) = f(z_i \mid \pi) f(y_i \mid z_i, \mu, \theta) \propto \prod_{i=1}^{n} \pi_{z_i} (\sigma_{z_i}^2)^{-1/2} e^{-\frac{1}{2\sigma^2}(y_i - \mu_{z_i})^2} \tag{1.9}$$

$$= \prod_{k=1}^{K} \pi_k^{n_k} (\sigma_k^2)^{-\frac{n_k}{2}} e^{-\frac{1}{2\sigma_k^2} \sum_{i:z_i=k} (y_i - \mu_k)^2} \tag{1.10}$$

which yields a posterior distribution with simple conditional distributions that can be sampled from very straightforwardly using a Gibbs sampler, in which the set of posterior variables is divided into blocks, each of which is iteratively sampled from its conditional posterior given the previous state of the others, which defines a Markov Chain whose stationary distribution is the full posterior (Geman and Geman, 1984).

1.4. Clustering Sequential Data: The Hidden Markov Model

In many applications, data are not i.i.d., but have some known dependence structure. Sequential data is one such case. Suppose a dataset consists of an ordered sequence of observations, $y_t, t = 1, \ldots, T$. Since unlike in the i.i.d. case we assume the order

is meaningful (most typically, the index t might represent the time at which the data point was collected), it is reasonable to expect that observing y_t would alter the predictive distribution at time $t+1$, even if we had access to the unconditional distribution of y_{t+1}. For example, y_t might be a word or a sentence of text in a document, a snippet of a sound recording, physiological measurements collected over time. In all of these cases, we expect nearby observations either to be similar, or to be made more predictable given the last few observations.

One option would be to encode the dependencies among the data points through a conditional model of the data at time t given various combinations of data at time $t-1$, $t-2$, ..., $t-s$; that is, to model the data using an order s Markov chain. However, modeling the distribution conditioned directly on observations risks an overly complex conditioning structure, since in most settings there is likely to be some idiosyncratic noise present in each observation that does not depend on time in the same way as the underlying signal. That is, quite often the distribution at t is likely to be a mixture of a temporally dependent component and a temporally independent component.

The **hidden Markov model** (HMM), first developed by Baum and Petrie (1966) (see also Rabiner and Juang (1986) or Rabiner (1989) for a classic tutorial) separates temporal dynamics of an evolving underlying ("hidden") state from an temporally independent "noise" component by augmenting the observed time sequence $\{y_t\}, t = 1, \ldots, T$ with a latent state sequence, $\{z_t\}, t = 1, \ldots T$, such that at the distribution of the observations, y_t are mutually independent conditioned on the latent sequence, z_t, but the z_t evolve according to a Markov chain. More formally, the model is

$$z_1 \sim \mathsf{Categorical}(\pi_{01}, \ldots, \pi_{0J}) \tag{1.11}$$

$$z_t \,|\, z_{t-1} \sim \mathsf{Categorical}(\pi_{z_{t-1}1}, \ldots, \pi_{z_{t-1}J}), \quad t = 2, \ldots, T \tag{1.12}$$

$$y_t \,|\, z_t \sim F(\theta_{z_t}), \quad t = 1, \ldots, T \tag{1.13}$$

where the vector $\pi_0 = (\pi_{01}, \ldots, \pi_{0J})$ is the initial distribution of the underlying

Markov chain, the matrix $(\pi_{jj'}), j, j' = 1, \ldots, J$ is the transition matrix of the Markov chain, F is a family of **emission distributions**, which is parameterized for each state in the Markov chain by the $\theta_j, j = 1, \ldots, J$.

This model is nearly identical to the mixture model discussed in Sec. 1.3, the only difference being the dependence of the label, z_t on the previous label, z_{t-1}. Though this may appear to be a small change, it implies that inferences about any particular latent state, z_t are potentially influenced by the entire data sequence, $y_1, \ldots, y_T$, and moreover, no latent state is independent of any other. Whereas in the static mixture model case, once the emission parameters, $\theta_1, \ldots, \theta_J$ and transition matrix $\boldsymbol{\pi} = (\pi_{jj'}), j, j' \in \{1, \ldots, J\}^2$ are known, finding the maximum *a posteriori* (MAP) set of class labels reduces to finding the MAP for each z_t separately, in the dynamic case, the best *marginal* z_t is not, in general, the tth element of the best *joint* sequence. Some inference goals we might want to solve for an HMM are

1. **Smoothing:** Finding the marginal distribution of z_t at some t given the observations $y_1, \ldots, y_T$ and a fixed set of model parameters $\theta_1, \ldots, \theta_J$ and $\boldsymbol{\pi}$.

2. **Decoding:** Finding the MAP state sequence $z_1, \ldots, z_T$ given the observations $y_1, \ldots, y_T$ and a fixed set of model parameters $\theta_1, \ldots, \theta_J$ and $\boldsymbol{\pi}$.

3. **Parameter Estimation:** Finding the "best" (by some criterion) values of the model parameters, $\theta_1, \ldots, \theta_J$ and $\boldsymbol{\pi}$ for fixed J.

4. **Model Selection:** Finding the "best" (by some criterion) degree of model complexity, J.

I briefly outline the classic algorithms for the first three of these in the next section; for more detail the reader is referred to any of a number of classic references, for example Rabiner and Juang (1986); Rabiner (1989); Bishop (2006). The model selection problem for static mixture models as well as the HMM is briefly introduced in Sec. 1.6 and treated in more detail in Chapter 2.

Smoothing We first consider the so-called "smoothing" problem; that is, finding the marginal posterior distribution over the value of the hidden state z_t corresponding to a particular observation y_t, having recorded the entire observation sequence, $y_1, \dots y_T$, which relative to z_t consists of the "past", $y_1, \dots, y_{t-1}$, the "present", y_t, and the future $y_{t+1}, \dots, y_T$.

For conciseness, define $z_{s:t} = z_s, \dots, z_t$. Then the full joint distribution is defined to be

$$p(z_{1:T}, y_{1:T}) = \prod_{t=1}^{T} p(z_t \mid z_{t-1}) p(y_t \mid z_t) \tag{1.14}$$

$$= \prod_{t=1}^{T} \pi_{z_{t-1}, z_t} f(y_t \mid \theta_{z_t}) \tag{1.15}$$

where for simplicity we define z_0 to be a special start state, $\pi_{0,j}, j = 1, \dots, J$ to be the initial distribution, and $f(\cdot \mid \theta_j)$ to be the emission PDF/PMF for state j.

Naively, we could compute $p(z_t \mid y_{1:T})$ by brute force, as

$$p(z_t \mid y_{1:T}) \propto p(z_t, y_{1:T}) = \sum_{z_1=1}^{J} \cdots \sum_{z_{t-1}=1}^{J} \sum_{z_{t+1}=1}^{J} \cdots \sum_{z_T=1}^{J} p(z_{1:t-1}, z_t, z_{t+1:T}, y_{1:T}), \tag{1.16}$$

but this has computational complexity which is exponential in T. Fortunately, we can take advantage of the Markov property that the past $z_{1:t-1}$ and the future $z_{t+1:T}$ are independent conditioned on the present to simplify computation considerably, using the factorization

$$p(z_t, y_{1:T}) = p(z_t, y_{1:t}) p(y_{t+1:T} \mid z_t). \tag{1.17}$$

The two terms on the right can be computed using the "forward" and "backward" recursion formulas:

$$p(z_t, y_{1:t}) = \sum_{z_{t-1}} p(z_{t-1}, y_{1:t-1}) p(z_t \mid z_{t-1}) p(y_t \mid z_t) \tag{1.18}$$

$$p(y_{t+1:T} \mid z_t) = \sum_{z_{t+1}} p(z_{t+1} \mid z_t) p(y_{t+1} \mid z_{t+1}) p(y_{t+2:T} \mid z_{t+1}) \tag{1.19}$$

where we define the initial conditions

$$p(z_0, y_0) \equiv 1 \tag{1.20}$$

$$p(y_{T+1:T} \mid z_t) \equiv 1 \tag{1.21}$$

The recursion formulas can be written more concisely in terms of matrix and vector operations by defining $\boldsymbol{\pi}$ to be the $J \times J$ transition matrix (i.e., $\boldsymbol{\pi}_{j,j'} = p(z_{t+1} = j' \mid z_t = j)$ for any t), $\mathbf{b}_t$ to be the $J \times 1$ likelihood vector with jth element defined to be $p(y_t \mid z_t = j)$, and $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ to be the forward and backward "message" vectors:

$$\alpha_{t,j} = p(z_t = j, y_{1:t}) \qquad \beta_{t,j} = p(y_{t+1:T} \mid z_t). \tag{1.22}$$

Then

$$p(z_t \mid y_{1:T}) = \frac{\boldsymbol{\alpha}_t \odot \boldsymbol{\beta}_t}{\boldsymbol{\alpha}_t^{\mathsf{T}} \boldsymbol{\beta}_t} \tag{1.23}$$

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}^{\mathsf{T}} \boldsymbol{\pi} \odot \mathbf{b}_t \tag{1.24}$$

$$\boldsymbol{\beta}_t = \boldsymbol{\pi}^{\mathsf{T}} (\mathbf{b}_{t+1} \odot \boldsymbol{\beta}_{t+1}) \tag{1.25}$$

where $\odot$ is the elementwise product. Taken together, this computation is known as the **forward-backward algorithm** (see also Rabiner and Juang (1986)).

Note that having found $p(z_T \mid y_{1:T})$, it is easy to find the predictive distribution for either z_{T+1} or y_{T+1}: simply write

$$p(z_{T+1} \mid y_{1:T}) = \sum_{z_T} p(z_T \mid y_{1:T}) p(z_{T+1} \mid z_T) \tag{1.26}$$

$$p(y_{T+1} \mid y_{1:T}) = \sum_{z_{T+1}} p(z_{T+1} \mid y_{1:T}) p(y_{T+1} \mid y_{z_{T+1}}) \tag{1.27}$$

repeating as often as needed to find $p(z_{T+s} \mid y_{1:T})$ or $p(y_{T+s} \mid y_{1:T})$ for $s > 1$.

Decoding Rather than finding the marginal distribution of some z_t, we may be interested in finding the most likely joint sequence, $z_{1:T}$, given all observations. This is known as the *decoding* problem. Like the smoothing problem, it could in principle be

solved by brute force, enumerating all possible sequences and calculating the posterior probability of each one, but again there is an efficient recursive solution, known as the **Viterbi algorithm**.

The idea is to develop a recursive formula to find $\hat{z}_{t+1:T,j}$, defined as the most likely "suffix" conditioned on $z_t = j$ and the data, $y_{1:T}$:

$$\hat{z}_{t+1:T,j} := \arg\max_{z_{t+1:T}} p(z_{t+1:T} \mid z_t = j, y_{1:T}). \tag{1.28}$$

By conditional independence of $z_{t+1:T}$ and $y_{1:t-1}$ given $z_t = j$, $p(z_{t+1:T} \mid z_t = j, y_{1:T})$ is proportional (as a function of $z_{t+1:T}$) to

$$p(z_{t+1:T}, y_{t:T} \mid z_t = j) = p(y_t \mid z_t = j)p(z_{t+1:T} \mid z_t)p(y_{t+1:T} \mid z_{t+1:T}) \tag{1.29}$$

and hence we can maximize this instead. Define $\mu_{t,j}$ to be the maximized probability itself:

$$\mu_{t,j} := \max_{z_{t+1:T}} p(z_{t+1:T}, y_{t:T} \mid z_t = j) \tag{1.30}$$

To develop the recursion relation, we take advantage of the fact that

$$\begin{aligned}
\mu_{t-1,j} &:= \max_{z_{t:T}} p(z_{t:T}, y_{t-1:T} \mid z_{t-1} = j) && (1.31)\\
&= \max_{j'} \max_{z_{t+1:T}} p(z_t = j', z_{t+1:T}, y_{1:T} \mid z_{t-1} = j) && (1.32)\\
&= \max_{j'} p(y_{t-1} \mid z_{t-1} = j)p(z_t = j' \mid z_{t-1} = j)p(\hat{z}_{t+1:T,j}, y_{t:T} \mid z_t = j') && (1.33)\\
&= \max_{j'} p(y_{t-1} \mid z_{t-1} = j)p(z_t = j' \mid z_{t-1} = j)\mu_{t,j'}. && (1.34)
\end{aligned}$$

where $\hat{z}_{t:T,j}$ is obtained by taking the j' that maximizes the above expression for each j, and prepending it to $\hat{z}_{t+1:T,j'}$. The initial condition in the recursion is simply the empty suffix for all j with $\mu_{T,j} \equiv 1$.

Thus at each step we start with J suffixes, one for each value of z_t, expand to J^2 by considering each possible value of z_{t-1} in combination with each suffix, and then pruning to J by taking advantage of the fact that, conditioned on z_{t-1}, the

information not yet considered (namely $z_{1:t-2}$ and $y_{1:t-1}$) is independent of the suffix. Once we have carried the recursion through to $t = 0$ (for which z_t is fixed at the special initial state), we have $\hat{z}_{1:T}$, which by definition is the MAP state sequence conditioned on all the data.

Sampling from $p(z_{1:T} \mid y_{1:T})$ The Viterbi algorithm deterministically yields the single best state sequence, but does not give us any measure of confidence in that result. If instead we wish to sample state sequences from the full joint posterior distribution over $z_{1:T}$, we can do so by first carrying out the forward part of the forward-backward algorithm, to yield $p(z_t \mid y_{1:T})$ for each $t = 1, \ldots, T$, sampling a value for z_T from its marginal posterior distribution, and then conditioning on this value to calculate

$$p(z_{T-1} \mid z_T, y_{1:T}) = p(z_{T-1} \mid z_T, y_{1:T-1}) \tag{1.35}$$

$$\propto p(z_{T-1} \mid y_{1:T}) p(z_T, \mid z_{T-1}) \tag{1.36}$$

where the first line follows from the conditional independence structure, and the second from Bayes' rule. We can iteratively apply this sampling method to obtain a sample of the full sequence $z_{1:T}$.

Estimating $\theta_1, \ldots, \theta_J$ **and** $\boldsymbol{\pi}$ The smoothing, prediction, decoding, and sampling algorithms given above assume that the transition matrix $\boldsymbol{\pi}$ and the emission parameters $\theta_1, \ldots, \theta_J$ are known. If, as is most often the case, we want to learn these parameters as well, we can do so. First, however, we need to define an estimation criterion. The two most common approaches are (1) to maximize the likelihood, $p(y_{1:T} \mid \theta_1, \ldots, \theta_J, \boldsymbol{\pi})$, which is a frequentist method, or to take a Bayesian perspective, putting a prior on the parameters, and working with the posterior distribution in some way (e.g., by taking samples from it using MCMC). In the When the priors are conditionally conjugate given $z_{1:T}$, a simple Gibbs sampling approach is to alternate

between sampling $z_{1:T}$ conditioned on $\theta_1, \ldots, \theta_J$ and $\boldsymbol{\pi}$, as described in the previous section, and sampling $\theta_1, \ldots, \theta_J, \boldsymbol{\pi}$ from their conditional posterior given $z_{1:T}$.

Alternatively, a local maximum of the likelihood can be obtained using Expectation-Maximization (EM). The augmented likelihood of $z_{1:T}$ and $y_{1:T}$ is

$$p(z_{1:T}, y_{1:T} \mid \boldsymbol{\pi}, \theta_1, \ldots, \theta_J) = \prod_{t=1}^{T} \pi_{z_{t-1} z_t} f(y_t \mid \theta_{z_t}) \tag{1.37}$$

$$= \prod_t \prod_j f(y_t \mid \theta_j)^{I(z_t=j)} \prod_{j'} \pi_{jj'}^{I(z_{t-1}=j', z_t=j)} \tag{1.38}$$

By replacing the indicator variables with their conditional expected values (given both the model parameters and the data) , denoting

$$q_{t,j} := \mathbb{E}[I(z_t = j \mid \boldsymbol{\pi}, \theta_1, \ldots, \theta_J, y_{1:T})] \tag{1.39}$$

$$\xi_{t,j,j'} := \mathbb{E}[I(z_{t-1} = j, z_t = j' \mid \boldsymbol{\pi}, \theta_1, \ldots, \theta_J, y_{1:T})] \tag{1.40}$$

we can perform the M step of the EM algorithm by maximizing

$$\prod_t \prod_j f(y_t \mid \theta_j)^{q_{t,j}} \prod_{j'} \pi_{jj'}^{\xi_{t,j,j'}} \tag{1.41}$$

To calculate the expectations for the E step, we first note that $q_{t,j} = p(z_t = j \mid y_{1:T})$, which we have from the forward-backward algorithm. Similarly, $\xi_{t,j,j'} = p(z_{t-1} = j, z_t = j' \mid y_{1:T})$, which we can get from the messages computed in the forward-backward algorithm by writing

$$\xi_{tjj'} \propto p(z_{t-1} = j, y_{1:t-1}, z_t = j', y_t, y_{t+1:T}) \tag{1.42}$$

$$= p(z_{t-1} = j, y_{1:t-1}) p(z_t = j' \mid z_{t-1} = j) p(y_t \mid z_t = j) p(y_{t+1:T} \mid z_t = j) \tag{1.43}$$

$$= \alpha_{t-1,j} \pi_{jj'} b_{tj'} \beta_{tj'} \tag{1.44}$$

1.6. Model Selection in HMMs and the Mixture of Gaussians Model

It is relatively rare that we would be in a situation where we wanted to fit a mixture of Gaussians to a dataset and knew exactly how many components there were in the

mixture. It is perhaps somewhat more likely that we know in advance how many states the hidden Markov chain in an HMM can visit, but there are certainly settings where we cannot or would not wish to do this. What can we do if we cannot specify K ahead of time?

One approach is to apply the logic outlined in Sec. 1.2.4 and simply put a prior distribution on K, and then calculate (or sample from) the posterior distribution over K. Then, although we can always achieve at least as large a likelihood with a larger K as with a smaller K (either by setting some mixture weights to zero, or by effectively combining multiple components into one by creating mixture components with identical means and variances), since models with larger K have more flexibility, their marginal likelihood will tend to be diluted by all of the ways they can fit the data badly.

However, this requires doing inference separately for as many values of K as we want to consider, and for the larger values of K, this inference can be computationally expensive. It turns out that we can achieve substantially the same thing more simply by employing a single **nonparametric** model with an *infinite* number of components.

In the next chapter, I examine the model selection problem in more detail, outlining the distinction between parametric and nonparametric models in the process. In the process, I develop some of the theory of **Dirichlet Process mixture models** (DPMM), which provides the theoretical foundation for an infinite-state hidden Markov model based on a hierarchical Dirichlet Process (the HDP-HMM). I conclude the next chapter by defining this model, giving additional detail on Gibbs sampling algorithms for both the static DPMM and HDP-HMM.

2. BAYESIAN NONPARAMETRIC MODELS AND THE INFINITE HIDDEN MARKOV MODEL

2.1. Parametric Vs. Nonparametric Models

Models that can be identified using a finite set of values (that is, parameters) are called **parametric** models. Their complexity and expressivity is the same whether they are fit using 10 data points, 10^6 data points, or 10^{10} data points. Most canonical statistical models are parametric: the parameters in a regression model comprise the regression coefficients and the parameters of the residual distribution. The parameters of the mixture of Gaussians model, for example, comprise a vector, π, of K mixing weights, as well as the parameters of the individual component Normal distributions: $\{\mu_k, \sigma_k^2\}, k = 1, \ldots, K$.

In a parametric model, once the sample size is a few orders of magnitude larger than the number of parameters, the gains made by further increasing the sample size, which in a frequentist setting comes in the form of narrower confidence sets, and in a Bayesian setting comes in the form of reduced posterior variance, are typically negligible: the errors in prediction due to inevitable model misspecification, and practical concerns with generalizability to new data sets, overwhelm the remaining decimal places of uncertainty.

A nonparametric model on the other hand, grows in complexity as the size of the dataset increases, attempting to capturing progressively more fine-grained structure. The name **nonparametric model** is something of a misnomer, in that nonparametric models do not have *no* parameters (a model with no parameters would by definition be unable to learn anything from data); rather, they have a number of parameters which grows adaptively as the sample size increases.

A common frequentist family of nonparametric models are kernel-smoothed den-

sity estimators (Rosenblatt et al., 1956; Parzen, 1962), in which the probability density of a data-generating process is estimated by taking the empirical distribution and "smoothing" it to obtain the estimated density at y by averaging together the values nearby points using a weight kernel. As a result, the estimated density requires n values to specify it, where n is the sample size. As such, the complexity of the family of distributions in the model space grows with the sample size, unlike in a parametric model. Here, the tradeoff between bias and variance is controlled not by the complexity of the model space, but rather by the choice of smoothing bandwith: how much of the density at each point is "borrowed" from nearby locations, with one extreme representing a bandwidth of zero, in which case the estimate is simply the empirical distribution, and the other extreme being an infinite bandwidth, in which case the density estimate is simply uniform, as all of the data is weighted equally at every location.

It is natural to interpret the kernel density estimator as a mixture model, where there is a mixture component centered at each data point whose distribution belongs to a family defined by the kernel function: for example, the Gaussian kernel results in a mixture of Gaussians, the Epanechnikov kernel (Epanechnikov, 1969) results in a mixture of Epanechnikov distributions, and the Uniform kernel results in a mixture of Uniform distributions. On the interpretation that each component in a mixture model represents a qualitatively distinct class of data, this means that each point is viewed as qualitatively distinct, which from a Bayesian perspective is unsatisfying. Instead, we would like a model whose complexity grows more slowly than linear in the sample size, such that as we collect additional data it is always possible to encounter something qualitatively new, but where the chances of doing so diminish as we have more and more data. This can be accomplished by employing a **Dirichlet Process** as the prior on the set of mixture components.

2.2. The Dirichlet Process

The Dirichlet Process (DP) was formally defined by Ferguson (1973) as a random probability measure, G, over a space $\mathcal{X}$ equipped with the sigma-algebra $\mathcal{A}$, with the defining property that, for any partition of $\mathcal{X}$ consisting of measureable sets, $A_1, A_2 \ldots, A_k$, the random distribution given by the probabilities

$$\{P(A_1), P(A_2), \ldots, P(A_k)\}$$

has a Dirichlet distribution.

A Dirichlet distribution is defined by a mean distribution, $\pi_1, \pi_2, \ldots, \pi_k; \sum_i \pi_i = 1$, and a concentration parameter, α, which acts as an inverse variance parameter: as α goes to infinity, draws from the Dirichlet distribution are distributions close to the mean with increasingly high probability.

The Dirichlet Process is also defined by a mean distribution G_0, called a **base measure**, and a concentration parameter α, but since the DP induces a Dirichlet distribution over *any* finite partition of $\mathcal{X}$, the mean distribution must be defined on all measureable sets in $\mathcal{A}$. We will write

$$G \sim \mathsf{DP}(\alpha G_0) \tag{2.1}$$

to indicate that the random measure G is distributed according to a Dirichlet Process with base measure G_0 and concentration parameter α. Then, concretely, we have

$$P(A_1), P(A_2), \ldots, P(A_k) \sim \mathsf{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_k))$$

An important probability of the DP is that the resulting measure is discrete almost surely, and hence it is a sensible choice as a prior on mixture components. An equally important property when it comes to using a DP as a Bayesian prior is that it is a *conjugate prior* to a discrete likelihood. That is, if n observations, $Y_1, \ldots, Y_n$, are drawn from some unkown discrete probability measure G, and the prior employed for

G is a Dirichlet Process, then the posterior distribution on G is also a DP, whose base measure is a weighted sum of the prior base measure, G_0, and the empirical distribution, $\hat{F}_n$:

$$G \,|\, Y \sim \alpha G_0 + n\hat{F}_n \tag{2.2}$$

and whose concentration parameter is simply $\alpha + n$.

2.2.1. The normalized Gamma Process representation of the DP

Ferguson (1973) also showed that the Dirichlet Process arises by normalizing a **Gamma Process**. A Gamma Process is a stochastic point process on $\mathbb{R}^+ \times \Theta$ which is defined by a **Lévy intensity measure**:

$$\nu(d\pi, d\theta) = \alpha\pi^{-1}e^{-\pi}d\pi G_0(d\theta) \tag{2.3}$$

where the measures of a subset, A of the domain is defined as the integral over A with respect to ν.

A realization of a Gamma process is a collection of point masses $\{\pi_k, \theta_k\}$, where $\pi_k \in \mathbb{R}^+$ is the mass associated with the point at location $\theta_k \in \Theta$. This collection can be used to define a measure, μ, on Θ, where

$$\mu(A) = \sum_{k:\theta_k \in A} \pi_k \tag{2.4}$$

The number $n(A)$ of point masses in a region $A \subset \mathbb{R}^+ \times \Theta$ is distributed as

$$n(A) \sim \mathsf{Poisson}(\int_A \nu(d\pi, d\theta)) \tag{2.5}$$

The Lévy intensity measure of the Gamma process satisfies conditions to guarantee that the sum of all of the π_k weights is finite with probability 1, and therefore the measure G defined by

$$G = \frac{\mu}{\sum_k \pi_k} \tag{2.6}$$

is a valid probability measure. Ferguson (1973) showed that this probability measure is a DP with base measure G_0 and concentration α, where these are the measure and parameter used in defining the Lévy intensity of the Gamma process.

Although the formal definition of the DP is well-defined, and although the Normalized Gamma Process representation guarantees existence of the DP, neither of these is terribly useful in *constructing* a DP, which limits the usefulness of the DP in applied modeling. Fortunately, a constructive definition of the DP was discovered by Sethuraman (1994), using what is known as a **Stick-Breaking Process**.

2.2.2. A stick-breaking process construction of the DP

As shown by Sethuraman (1994), we can generate a draw from a Dirichlet Process by iteratively sampling the π_k weights, and then placing a point mass with weight π_k at a location in Θ independently drawn from G_0. This process is called a **Stick-Breaking Process**. By using the following algorithm to select the stick weights, the resulting collection of point masses has a Dirichlet Process.

Having drawn $k-1$ point masses $(\pi_1, \theta_1), \ldots, (\pi_{k-1}, \theta_{k-1})$,

1. Draw $\tilde{\pi}_k \sim \mathsf{Beta}(1, \alpha)$.
2. Set $\pi_k = \tilde{\pi}_k \prod_{k'=1}^{K-1}(1 - \pi_{k'})$.
3. Draw $\theta_k \sim G_0$.

where, when $k = 1$, the null product in step 2 is 1.

The choice of G_0 and α determine the resulting DP.

When describing the stick-breaking part of this process by itself — that is, the process that produces the weights, $\pi_1, \pi_2, \ldots$, it is common to write

$$\pi \sim \mathsf{GEM}(\alpha) \tag{2.7}$$

where GEM stands for Griffiths-Engen-McCloskey, the names of three authors who did early work on stick-breaking processes and laid the foundation for the connection to Dirichlet Processes later formalized by Sethuraman (1994).

2.2.3. The Chinese Restaurant Process

The Dirichlet Process commonly acts as a prior distribution on the parameters of an infinite mixture model, where the parameters consist of the infinite set of mixing weights, $\boldsymbol{\pi} := \pi_1, \pi_2, \ldots$, and the infinite set of component parameters, $\boldsymbol{\theta} := \theta_1, \theta_2, \ldots$. Ultimately what we generally care about is to make inferences about these parameters, or, if the goal is clustering, a set of indicator variables, $z_1, \ldots, z_N$ associating each data point with a mixture component. Since there are infinitely many parameters in the model we cannot hope to estimate all of them explicitly. Several solutions are available. First, one can truncate the model, representing only a finite number, J, of components. Alternatively, one might explicitly represent only the parameters of those components associated with data points (of which there are necessarily finitely many), as well as one aggregated "unobserved" component. However, in many applications it is not necessary to know the component parameters themselves, only the cluster assignments. In this case, we can work with the marginal distribution of $z_1, \ldots, z_N$, having integrated out the mixing weights $\boldsymbol{\pi}$, which can be done analytically. The process of drawing samples from this marginal distribution is often described in terms of a metaphor of customers (which represent the data points) sharing food (which represents the mixture components) at a Chinese restaurant. The metaphor usually goes something like the following.

One by one, customers enter a Chinese restaurant and either sit at an unoccupied table and order a dish for the table, or join a table with other customers and share whatever dish is at the table. The first customer necessarily starts a new table. Subsequent customers join table k with probability proportional to n_k, where n_k is

the number of customers currently seated at that table, and sit at an unoccupied table with probability proportional to a parameter α.

More formally, the $n+1$th customer to enter the restaurant is assigned to table k according to the distribution

$$P(t_{n+1} = k) = \begin{cases} \frac{n_k}{n+\alpha} & k = 1, \ldots, K \\ \frac{\alpha}{n+\alpha} & k = K+1 \end{cases} \tag{2.8}$$

For each new table, a dish, θ_k is sampled from a base distribution, G_0.

It turns out that the distribution of the collection of assignments of dishes to customers is the same as the *marginal* distribution of draws from a random measure G which is distributed $\mathsf{DP}(\alpha G_0)$ (see Teh (2011) for a proof of this fact as well as a review of the theory of DPs in general).

Notice that the Chinese Restaurant Process (CRP) has the property that the more often a dish has already been selected, the more likely it is to be selected again: that is, it has a "rich get richer" quality. This makes sense in terms of the marginal distribution of draws from a DP-distributed random measure, since the more observations there are at a particular location, the stronger the evidence that the mass under G at that location is large, and hence, the larger the posterior predictive probability at that location (marginalizing over G).

2.2.4. The Dirichlet Process mixture model

We are now ready to define a nonparametric version of a Bayesian Gaussian Mixture Model which replaces the Dirichlet distribution prior on the collection of mixture components from the fixed K mixture model with a Dirichlet Process prior.

Reiterating the model initially defined in (1.4), suppose we have a model of the form

$$f(y \mid \pi, \theta) = \sum_{k=1}^{\infty} \pi_k f(y \mid \theta_k) \tag{2.9}$$

where $\theta = (\theta_1, \theta_2, \dots)$ are some parameters governing the mixture components, and $\pi = (\pi_1, \pi_2, \dots)$ is the vector of mixing weights that sum to 1. We now drop the assumption that all but finitely many of the π_k are zero, and instead adopt as a prior:

$$\{(\pi_k, \theta_k)_{k=1}^{\infty}\} \sim \mathsf{DP}(\alpha G_0) \tag{2.10}$$

As before, we introduce indicator variables, $\{z_i\}_{i=1}^n$ so that we can write

$$P(z_i = k) = \pi_k, \qquad i = 1, \dots, n; k = 1, 2, \dots \tag{2.11}$$

$$y_i \,|\, z_i \sim F(\theta_{z_i}) \tag{2.12}$$

where F is a parametric family parameterized by θ. For example, if F is Normal, then θ might represent the mean vector and covariance matrix, and we might choose G_0 to be a Normal-Inverse Wishart distribution (or a non-conjugate choice as the application suggests).

2.2.5. Two Gibbs samplers for DP mixture models

Given data, $y = (y_1, \dots, y_n)$ modeled using the mixture model in (2.9) with the prior in (2.10), we want to be able to make inferences about the parameters, (π_k, θ_k). As with all but the simplest models, the full posterior is not amenable to exact analysis, and so we must resort to approximation to compute quantities of interest. There are two dominant approximation methods in Bayesian inference. One is variational inference, in which the true posterior is replaced by a distribution which is more amenable to analysis and which is obtained by selecting *a priori* a family of tractable distributions, and then minimizing some measure of discrepancy between the chosen distribution and the true posterior (typically the objective is to minimize the KL divergence from the true posterior to the estimate). The second popular approximation technique is Markov Chain Monte Carlo (MCMC), in which integrals involving the posterior are computed based on a sample from the posterior drawn using a Markov

Chain whose transition kernel is chosen so as to yield the true posterior (as determined by the data and the model) as the unique stationary distribution. In a nutshell, variational Bayes provides an exact calculation based on an approximate distribution, whereas MCMC provides approximate calculations based on the true posterior. I will focus on MCMC in this dissertation. See Fox and Roberts (2012) for a tutorial on variational Bayes generally and Blei and Jordan (2006) and Kurihara et al. (2007) for work on variational methods for Dirichlet Process mixture models in particular.

As with the finite mixture model, MCMC inference is greatly simplified by sampling over the z_i indicators as well as over the mixture component parameters themselves (indeed, in some applications, these indicators may be the variables of primary interest).

We are interested in the joint posterior over the emission parameters, $\{\theta_k\}$, the component weights, $\{\pi_k\}$, and the component indicators, $\{z_i\}$, where k indexes components and i indexes observations.

Method 1: A Collapsed Sampler Based on the CRP One option is to sample only θ and z, integrating out π. This is possible since the marginal distribution of $z \,|\, \theta$ is a Chinese Restaurant Process.

In this approach, we sample each entry of z one at a time conditioned on the others. Assume we have an initial assignment of data points to components. Let $z^{(-i)}$ be the vector of component indicators excluding z_i, let $n_k^{(-i)} = \sum_{i' \neq i} \mathbb{I}(z_i = k)$ be the count of the number of $z_{i'}$ currently assigned to component k, not counting z_i, and let K count the number of distinct values in z among the other $n-1$ indicators (we will assume by permuting the labels that the distinct values are numbered 1 through K). Then we have

$$p(z_i = k \,|\, z^{(-i)}) = \begin{cases} \frac{n_k}{n+\alpha} & k = 1, \ldots, K \\ \frac{\alpha}{n+\alpha} & k = K+1 \end{cases} \tag{2.13}$$

where if z_i takes on a distinct value from any of the other entries in z, we call this

value $K+1$.

In order to sample z_i conditioned on both $z^{(-i)}$ and the data, we also need to compute the likelihood $p(y_i \mid z_i, \theta)$ for each value of z_i. Generically, we simply have

$$p(y_i \mid z_i, \theta) = f(y_i \mid \theta_{z_i}) \tag{2.14}$$

where $f(\cdot \mid \theta_{z_i})$ is the density (or mass) function corresponding to the distribution $F(\theta_{z_i})$.

Thus we sample z_i with probabilities

$$p(z_i = k \mid z^{(-i)}, y_i, \theta) \propto \begin{cases} \frac{n_k}{n+\alpha} f(y_i \mid \theta_k) & k = 1, \ldots, K \\ \frac{\alpha}{n+\alpha} \int_\Theta f(y_i \mid \theta_*) p(\theta_*)\, d\theta_* & k = K+1 \end{cases} \tag{2.15}$$

where depending on the form of f, it may be possible to compute the integral for the marginal likelihood of y_i analytically; but if not, we can approximate it by sampling a value of θ_* from the prior and using

$$p(z_i = K+1 \mid \theta, y) \propto \frac{\alpha}{n+\alpha} f(y_i \mid theta_*) \tag{2.16}$$

If z_i is set to $K+1$, we increment K. If z_i was previously the only indicator assigned to some value k, such that now cluster k has no data associated with it, relabel the indicators to occupy consecutive natural numbers (this step is purely for computational convenience, as the labels represent categories with no quantitative interpretation).

Conditioned on z, sampling θ amounts to K independent updates of the parameters of the K separate components, each using the likelihood of the observations currently assigned to that component. If the prior on θ is conjugate to the likelihood, this involves K samples from an exponential family.

Method 2: An uncollapsed sampler based on the Stick-Breaking Process Alternatively we might choose to sample a finite subset of the entries in $\boldsymbol{\pi}$ directly.

Conditioned on the current set of assignments to the z indicators, we again let K represent the number of distinct values taken on by the z indicators, again assuming

that the labels have been re-numbered as needed to take the values 1 through K, and again let $n_k, k = 1, \ldots, K$ count how many observations are assigned to each k. We can then sample the weights associated with each currently represented component, given by $\pi_1, \ldots, \pi_K$, as well as the total weight associated with all unrepresented components combined, π_{new}. Given the n_k, we have

$$(\pi_1, \ldots, \pi_K, \pi_{new}) \sim \mathsf{Dirichlet}(n_1, n_2, \ldots, n_K, \alpha) \tag{2.17}$$

We may then sample each z_i according to

$$p(z_i = k \,|\, \pi, \theta, y) \propto \begin{cases} \pi_k f(y_i \,|\, \theta_k) & k = 1, \ldots, K \\ \pi_{new} \int_\Theta f(y_i \,|\, \theta_*) p(\theta_*) \, d\theta_* & k > K \end{cases} \tag{2.18}$$

where each time some z_i is assigned to a new component, we must instantiate a new π_{K+1} using the stick-breaking process, by sampling

$$\tilde{\pi}_{K+1} \sim \mathsf{Beta}(1, \alpha), \tag{2.19}$$

set $\pi_{K+1} := \pi_{new}\tilde{\pi}_{K+1}$, and set $\pi_{new} := 1 - \sum_{k=1}^{K+1} \pi_k$, and then increment K before sampling the next z_i.

Sampling θ is exactly as in the collapsed sampler, since the distribution depends only on $\mathbf{z}$ and $\mathbf{y}$.

2.3. An Infinite-State HMM

We would like to adapt the nonparametric DP mixture model to the sequential setting to define an infinite state HMM, analogous to the way that the finite mixture model was adapted to create a finite state HMM. There, the link between the mixture model and the J-state HMM was that the HMM consisted of a separate J-state mixture model associated with each state, for a total of J mixtures. That is, after visiting state j, the next observation is drawn from mixture model j.

Naively, then, we might try to define a countably infinite set of mixture models, each with a DP prior, and after visiting component j, draw the next observation from

mixture model j. However, if the prior on the emission parameters θ is continuous, then with probability 1 the set of mixture components will be non-overlapping, and hence we will never revisit the same component twice.

2.3.1. The hierarchical Dirichlet Process

The key property that we want in an Infinite State HMM is that the mixture distributions need to be coupled — that is, they need to share a set of components. In order to accomplish this using a Dirichlet Process prior on the mixture parameters, the base measure needs to be discrete.

2.3.2. The HDP-HMM

Teh et al. (2006) defined the **Hierarchical Dirichlet Process** (HDP), in which a collection of Dirichlet Processes take as a common base measure a distribution itself drawn from a DP. This model is then hierarchical in the same sense as a Bayesian hierarchical regression model, in which data sets from a collection of sources are assumed to have similar distributions, and hence are given a common prior whose parameters are informed by all of the data across sources.

Formally, we define

$$G_0 \sim \mathsf{DP}(\gamma H) \tag{2.20}$$

where H is some measure over a space Θ (in the mixture modeling context, Θ is the space of component parameters) and

$$G_j \sim \mathsf{DP}(\alpha G_0), \quad j = 1, 2, \ldots, J \tag{2.21}$$

where γ is a concentration parameter that governs the distribution of the weights of the atoms in the shared base measure G_0, with larger γ leading to probability mass being dispersed among a large number of atoms, and smaller γ leading to a

concentration of mass in just a few components. The concentration parameter α for the individual measures G_j governs how tightly clustered the individual G_j are around the mean base measure G_0, with large α leading to the G_j being highly similar to G_0, and small α leading to each G_j placing a lot of mass in a small number of components drawn from G_0, such that the average across G_js still looks like G_0, but where each G_j may be quite different. In practice a single α is usually shared across all j, but in principle different values could be used. The effect of α and γ on the characteristics of typical transition matrices sampled from the prior is illustrated in Fig. 2.1.

While the effect of γ is clear enough from the Stick-Breaking Process representation of DPs, it is worth examining how it is that the second-level concentration α governs the behavior of the atoms in the individual G_j. We can see this using the Stick-Breaking Process as well.

The effect of α is clearest when γ is large, so that G_0 has little bits of mass at lots of different locations in Θ, where those locations are distributed according to H. To draw a G_j with this discrete G_0 as a base measure, we begin with a stick of unit mass, and break off a mass by sampling from a $\mathsf{Beta}(1, \alpha)$ distribution. We place a mass with this weight at a random location drawn from the countable set of possibilities given by G_0. We then draw a random breaking point for remaining mass from another $\mathsf{Beta}(1, \alpha)$ distribution, choose another location independently from G_0, and so on. First, imagine α is quite small. Then the first mass is likely to be quite near 1, the secon is likely to take a large share of what's left, and so on. So G_j is likely to place most of its mass at a few locations, though *which* locations those are is uncertain due to the high dispersion in G_0. At the other extreme, suppose α is quite large. Then each time we break off a mass, it is likely to be quite small, and it will take many breaks and samples from G_0 before we accumulate significant probability. As a result, by the law of large numbers, the distribution of this large number of small masses is likely to mimic closely the distribution in G_0.

2.3.3. A stick-breaking representation of the aggregate weights in an HDP

The second-level Stick-Breaking Process describes the weights of the atoms drawn from the base measure to form each G_j; however, since the base measure is discrete so that the probability of assigning a stick to each specific location, θ_j is positive (and constant), and since infinitely many sticks are being placed, we will (with probability 1) revisit each location infinitely many times. Thus if we want to describe the total mass that G_j assigns to the location θ_k, we need to account for the fact that this mass is the accumulation of infinitely many stick masses. We can get a better handle on the distribution of these masses by considering the defining property of a Dirichlet Process random measure: namely, that the distribution of the values of the measure over any finite partition of Θ is a Dirichlet distribution.

Denote by $\theta_1, \theta_2, \ldots$ the locations of the atoms in G_0, with associated probabilities $\beta_1, \beta_2, \ldots$. Consider the finite partition of Θ into just two sets: $\{\theta_1\}$ and everything else (that is, $\Theta \setminus \{\theta_1\}$), and define $\pi_{jk} := G_j(\theta_k), k = 1, 2, \ldots$. By the definition of a DP, the random measures of these two sets have a Dirichlet distribution:

$$(\pi_{j1}, 1 - \pi_{j1}) \sim \mathsf{Dirichlet}(\alpha G_0(\theta_1), \alpha G_0(\Theta \setminus \theta_1)), \tag{2.22}$$

that is,

$$(\pi_{j1}, 1 - \pi_{j1}) \sim \mathsf{Dirichlet}(\alpha\beta_1, \alpha(1 - \beta_1)), \tag{2.23}$$

Since the first component of a two-component Dirichlet distribution has a Beta distribution with the same parameters, this means that $\pi_{j1} \sim \mathsf{Beta}(\alpha\beta_1, \alpha(1 - \beta_1))$.

More generally, for any K, we have

$$(\pi_{j1}, \ldots, \pi_{jK}, \sum_{k=K+1}^{\infty} \pi_{jk}) \sim \mathsf{Dirichlet}(\alpha\beta_1, \ldots, \alpha\beta_K, \alpha \sum_{k=K+1}^{\infty} \beta_k) \tag{2.24}$$

Thus we can construct the π_{jk} weights directly via the following Stick-Breaking Process. For $k = 1, 2, \ldots,$

1. Draw $\tilde{\pi}_{jk} \sim \mathsf{Beta}(\alpha\beta_k, \alpha(1 - \sum_{k'=1}^{k} \beta_{k'}))$.

2. Set $\pi_{jk} = \tilde{\pi}_k \prod_{k'=1}^{K-1}(1 - \pi_{k'})$.

2.3.4. A normalized Gamma Process representation of the weights in the HDP

A Dirichlet distribution can be constructed by normalizing a set of Gamma random variables, where the shape parameters are equal to the parameters of the Dirichlet, and the rate parameters are a constant (what constant does not matter, since it will be normalized out anyway). So we can write

$$\beta \sim \mathsf{GEM}(\gamma) \tag{2.25}$$

$$\tilde{\pi}_{jk} \overset{ind}{\sim} \mathsf{Gamma}(\alpha\beta_k, 1) \tag{2.26}$$

$$T_j := \sum_{k'} \tilde{\pi}_{jk'} \tag{2.27}$$

$$\pi_{jk} := T_j^{-1} \tilde{\pi}_{jk} \tag{2.28}$$

Since the sum of independent Gamma variates with a common rate parameter is a Gamma variate with the shared rate and whose shape is the sum of the shapes, we have $T_j \sim \mathsf{Gamma}(\alpha, 1)$, and conditioned on T_j the π_{jk} are independent.

2.3.5. Adapting the HDP for an infinite-state HMM

We could use the HDP to define a coupled collection of mixture models, to model clustered data in several known contexts, as Teh et al. (2006) did to define coupled infinite mixtures of topics in various documents. We can also use it to define an infinite collection of infinite mixtures in the form of an HMM with infinitely many states. Beal et al. (2001) first described an infinite HMM without explicitly making the connection to a Hierarchical Dirichlet Process; Teh et al. (2006) showed that, with a few differences, the model developed by Beal et al. could be derived using an HDP.

In the HMM setting, the "data sources" indexed by j in the notation in the previous section correspond to different previous states in the hidden Markov chain, θ_j represents the emission parameters associated with state j each hidden state, and G_j represents the transition distribution from state j to all other states (which are identified by their respective $\theta_{j'}$, and due to discreteness of G_0, are the same countably infinite set for each source state). We will denote by $\pi_{jj'}$ the transition probability from state j to state j', where we have replaced the k subscript by j' to emphasize the fact that the set of source states and the set of destination states are the same.

Then we can define the following prior on the elements of the transition matrix of the HDP-HMM:

$$\beta \sim \mathsf{GEM}(\gamma) \tag{2.29}$$

$$\tilde{\pi}_{jj'} \sim \mathsf{Gamma}(\alpha\beta_{j'}, 1) \tag{2.30}$$

$$T_j := \sum_{j'} \tilde{\pi}_{jj'} \tag{2.31}$$

$$\pi_{jj'} := T_j^{-1}\tilde{\pi}_{jj'} \tag{2.32}$$

To complete the model, define a Markov chain over state indicators, $\{z_t\}_{t=1}^T$, with infinite transition matrix $\pi = (\pi_{jj'})$, a prior measure, H, on the collection of emission parameters $\{\theta_j\}_{j=1}^\infty$, and the likelihood F:

$$z_t \,|\, z_{t-1} \sim \mathsf{Discrete}(\pi_{z_{t-1}1}, \pi_{z_{t-1}2}, \dots) \tag{2.33}$$

$$\theta_j \overset{i.i.d.}{\sim} H \tag{2.34}$$

$$y_t \,|\, z_t \sim F(\theta_j) \tag{2.35}$$

2.4. Inference in Finite and Infinite State HMMs

A variety of inference algorithms have been developed for both finite state and infinite state Bayesian Hidden Markov Models, including variational methods (see Beal (2003)

for a review of the finite-state case, and Johnson and Willsky (2014) for the infinite-state case), and particle filters (Fearnhead and Clifford, 2003; Tripuraneni et al., 2015), as well as a number of different Gibbs-sampling algorithms (Teh et al., 2006; Van Gael et al., 2008; Fox et al., 2008; Johnson and Willsky, 2013). The key difference among the Gibbs samplers is the treatment of the latent state sequence, $\{z_t\}$. One distinction is whether each z_t is put in its own block and sampled conditioned on all of the others (as in Teh et al.), or whether the full z_t sequence is sampled at once, as in Van Gael et al., Fox et al. and Johnson and Willsky. These two approaches trade off computational complexity per iteration with the degree of autocorrelation between successive samples. I focus here on the latter case, since the Gibbs sampler developed in Chapter 3 for the new HaMMLeT model is of this type.

2.4.1. The forward-backward algorithm

In a non-dynamic mixture model, the component labels are mutually independent given the component parameters, and hence a Gibbs sampler can trivially sample all of them simultaneously. In the dynamic case, however, the distribution of each state indicator depends on the previous indicator, which depends on the previous one, etc. Moreoever, as a result of this propagation of dependence, the data at time t is indirect evidence for the indicators not just at time t, but at *all other* times as well. As a result, sampling the $\{z_t\}$ sequence jointly requires care to appropriately account for all of these dependencies. The total number of possible sequences in a model with J states and T observations is T^J, which is an unmanageable number to consider exhausitvely. In this section I derive the **forward-filtering backward-sampling** algorithm for a J state HMM. This algorithm is closely related to the forward-filtering backward-smoothing (aka, simply "forward backward") algorithm described in section 1.5 of Chapter 1, which I summarize again here for convenience.

The forward-backward algorithm is an efficient dynamic-programming technique

for sampling the $\{z_t\}$ sequence given the transition matrix, π, the emission parameters, θ, and the data, $y_1, \ldots, y_T$, which requires only $O(J^2T)$ computation.

Because of the Markov assumption, the set $\{z_1, \ldots, z_{t-1}\}$ is conditionally independent of the set $\{z_{t+1}, \ldots, z_T\}$, as well as of the data $\{y_{t+1}, \ldots, y_T\}$, given the indicator z_t. The forward-backward algorithm takes advantage of this conditional independence property to iteratively compute the marginal distribution of each z_t given only the data from y_1 through y_t (the "forward step"), and then sample the $\{z_t\}$ sequence starting with z_T and working backward to z_1 (the backward step).

I describe the backward step first, since this will make clearer the need for the forward step.

The Backward Step In the backward step, we iteratively sample each z_t conditioned on its temporal successor z_{t+1} and on the model parameters and data. As the name implies, we do this starting from z_T and proceeding backward to z_1. As noted above, conditioned on z_{t+1}, z_t is conditionally independent of all "future" data: $y_{t+1}, \ldots, y_T$. Using this fact and Bayes' rule, we have

$$p(z_t = j \mid z_{t+1}, y) \propto p(z_t = j \mid z_{t+1}, y_1, \ldots, y_t) \tag{2.36}$$

$$\propto p(z_t = j \mid y_1, \ldots, y_t) p(z_{t+1} \mid z_t = j) \tag{2.37}$$

$$= b_t(j) \pi_{j z_{t+1}} \tag{2.38}$$

where $b_t(j)$ is shorthand notation for the probability $p(z_t = j \mid y_1, \ldots, y_t)$, and we have omitted dependence on π and θ for conciseness. Hence if we can compute $b_t(j)$ for each $j = 1, \ldots, J$ and each $t = 1, \ldots, T$, then it is straigtforward to sample the z_t sequence backward from T to 1. The computation of b_t is the purpose of the forward step.

The Forward Step The goal of the forward step is to compute at each t the partial posterior distribution b_t as defined above. We can accomplish this via an iterative

"message passing" scheme, beginning at b_1.

By definition of the HMM we have

$$p(z_1 = j \,|\, y_1) \propto p(z_t = j)p(y_1 \,|\, z_t = j) \tag{2.39}$$

$$= \pi_{0j} f(y_1 \,|\, \theta_j) \tag{2.40}$$

Now, suppose we have computed b_t for some t. Then we can calculate the next "message", b_{t+1} as follows, using the conditional independence properties of the HMM:

$$b_{t+1}(j') = p(z_{t+1} = j' \,|\, y_1, \ldots, y_{t+1}) \tag{2.41}$$

$$\propto p(z_{t+1} = j', y_1, \ldots, y_{t+1}) \tag{2.42}$$

$$\propto p(z_{t+1} = j', y_1, \ldots, y_t)p(y_{t+1} \,|\, z_{t+1} = j') \tag{2.43}$$

$$= \sum_j p(z_t = j, z_{t+1} = j', y_1, \ldots, y_t)p(y_{t+1} \,|\, z_{t+1} = j') \tag{2.44}$$

$$= \sum_j p(z_t = j, y_1, \ldots, y_t)p(z_{t+1} = j' \,|\, z_t = j)p(y_{t+1} \,|\, z_{t+1} = j') \tag{2.45}$$

$$\propto \sum_j b_t(j)\pi_{jj'} f(y_{t+1} \,|\, \theta_{j'}) \tag{2.46}$$

Thus to sample the full sequence $\{z_t\}_{t=1}^T$ we first do a forward pass to calculate $b_t(j)$ for each t, j, and then iteratively sample each z_t beginning from z_T and working backward to z_1. At each t we need to compute J^2 products and a sum over J terms, and this needs to occur for each $t = 1, \ldots, T$, so the overall computation required for the forward step is $O(TJ^2)$. The backward step requires only TJ multiplications, and so the full forward-backward algorithm is $O(TJ^2)$.

2.4.2. Gibbs sampling in the HDP-HMM

In a finite-state HMM we can construct a straightforward Gibbs sampler, employing the forward-backward algorithm to sample the state sequence, $\{z_t\}$, conditioned on the data and model parameters, and then conditioned on the state sequence, we can update the model parameters as in a non-dynamic mixture model: if conjugate priors

are used, this can be done exactly; otherwise a proposal distribution can be chosen and the result accepted according to the Metropolis-Hastings acceptance probability formula.

In the HDP-HMM, however, we have infinitely many states to consider, and so we cannot directly apply the forward-backward algorithm to sample the state sequence, since it requires evaluating the probability of being in every state at every time. Thus if we want to construct a Gibbs sampler for the HDP-HMM, we have a few obvious options. One is to sample each z_t separately conditioned on the rest of the sequence, integrating out the transition matrix. This is the approach taken in the Chinese Restaurant Franchise representation, which also integrates out the top level weights, β, as well as in the related auxiliary variable sampler, which instantiates β, both originally described in Teh et al. (2006) (see that paper for details). Alternatively, we can use a finite-state approximation, either by adaptively terminating the stick-breaking process once a target proportion, $(1-\varepsilon)$, for some suitably small choice of ϵ, of the stick has been broken off (this is referred to as a **truncated approximation**), or by simply picking a fixed reasonably large but finite J. This last approach is pursued by Fox et al. (2008) and Johnson and Willsky (2013) in their generalizations of the HDP-HMM: the Sticky HDP-HMM and the HDP Hidden Semi-Markov Model (HDP-HSMM), respectively, and is used in experiments presented in this dissertation using the HDP-HMM-LT model which is defined in Chapter 3.

2.5. Local Transitions

HDP models have been widely adopted in Bayesian statistics and machine learning. However, a limitation of the vanilla HDP is that it offers no mechanism to model correlations between mixture components across contexts. This is limiting in many applications, where we expect certain components to occur or not occur together. In the HMM setting, we might expect certain states to exhibit similar incoming

and outgoing transition probabilities; that is, for certain rows and columns of the transition matrix to be correlated. In particular, we might expect pairs of states that are "similar" in some way to transition frequently to each other. The HDP-HMM offers no mechanism to model this similarity structure.

In Chapter 3, I define a new model, the Hierarchical Dirichlet Process Hidden Markov Model with Local Transitions (HDP-HMM-LT, or HaMMLet, for short), which allows for correlations between rows and columns of the transition matrix by assigning each state a location in an abstract metric space and promoting transitions between states that are near each other. I present a Gibbs sampling scheme for inference in this model which employs an auxiliary variable scheme to produce simple conditional distributions. Then, in Chapters 4 and 5 I define concrete instantiations of this model for several kinds of data, and report experimental results showing that the new model in some cases achieves better inference performance than existing models.

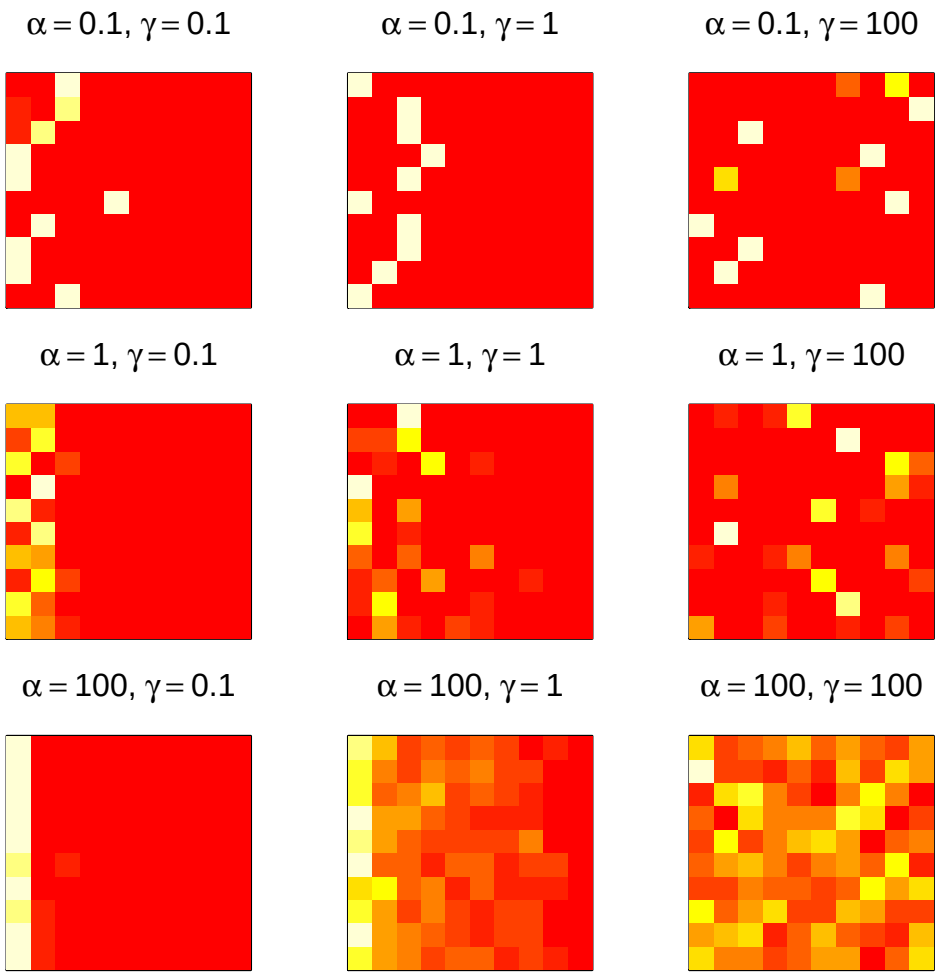


Figure 2.1: Illustration of the effect of α and γ concentration parameters on the properties of a typical 10-state transition matrix sampled from the HDP prior with top level concentration parameter γ and second level concentration parameter α. In the left column, when γ is small, most of the mass in each row is allocated to a small number of states; as γ increases, more states are used. In the top row, when α is small, the transition probabilities are more dependent on the previous state, with each state tending to have a small number of "preferred" states that are visited next. In contrast, as α increases, the rows of the transition matrix become more similar, closely reflecting the top level distribution, and so the successive state labels are closer to being independent.

3. HAMMLET: AN INFINITE HIDDEN MARKOV MODEL WITH LOCAL TRANSITIONS

In this chapter, I describe a generalization of the Hierarchical Dirichlet Process Hidden Markov Model Teh et al. (2006) discussed in Chapter 2 which introduces a notion of latent similarity between pairs of hidden states, such that transitions are a priori more likely to occur between states that are deemed "similar". This is achieved by placing a similarity function on the space of state parameters which returns for each pair of states a value between 0 and 1, representing the degree to which they are similar (in whatever sense is desired for the application at hand), and scaling HDP-generated transition probabilities by the similarity between states. I will refer to this model as the Hierarchical Dirichlet Process Hidden Markov Model with Local Transitions (HDP-HMM-LT, or HaMMLeT). Although this achieves the goal of selectively increasing the probability of transitions between similar states, inference is made more complicated since, unlike in the "vanilla" HDP-HMM, the posterior measure over transition distributions is no longer a Dirichlet Process — that is, the prior is no longer conjugate to the state sequence likelihood.

I will present an alternative representation of this process that facilitates inference with an auxiliary variable scheme with the following interpretation: The discrete time chain is recast as a continuous time Markov process in which: (1) some jump attempts fail, (2) the probability of success is proportional to the similarity between the source and destination states, (3) only successful jumps are observed, and (4) the time elapsed between jumps, as well as the number of unsuccessful jump attempts, are latent variables that are sampled during MCMC inference. By introducing these auxiliary latent variables, nearly all conditional distributions in the model are members of an exponential family, admitting exact Gibbs sampling. The only exception is the set of similarities, which are defined in an application-specific way, and require

application-specific inference methods, which may or may not admit sampling directly from the conditional posterior.

One class of application in which it is useful to incorporate a notion of locality occurs when the latent state sequence consists of several parallel chains, so that the global state changes incrementally, but where these increments are not independent across chains. Factorial HMMs Ghahramani et al. (1997) are commonly used in this setting, but this ignores dependence among chains, and hence may do poorly when some combinations of states are much more probable than suggested by the chain-wise dynamics. The application that originally motivated this model is in this category, and involves the extraction of multivariate scope information from the description of biological processes in natural language text. However, in the interest of clearer illustration of the model's properties without the need for the complex and highly structured feature sets that are inevitably involved in NLP, I will instead focus on the simpler setting of blind source separation in the context of speaker diarization. This application is described in detail in Chapter 4.

Another setting where the LT property is useful is when there is a notion of state geometry that licenses syllogisms: e.g., if A frequently leads to B and C and B frequently leads to D and E, then it may be sensible to infer that A and C may lead to D and E as well. This property is arguably present in musical harmony, where consecutive chords are often (near-)neighbors in the "circle of fifths", and small steps along the circle are more common than large ones. I evaluate the model's ability to learn harmonic structure from unlabeled (synthetic and real) music data in Chapter 5.

In the remainder of this chapter, I first review the transition dynamics in the "vanilla" HDP-HMM so that it is easier to see how the proposed HDP-HMM-LT differs; I then define the generative process of the HDP-HMM-LT; finally, I introduce the "Markov Jump Process With Failed Transitions" augmented data representation, which gives rise to a natural Gibbs sampler for the HDP-HMM-LT model.

Notational Conventions In the definitions and derivations that follow, I will adopt the convention that variables written with no subscript, such as θ, β, and π, represent the collection of all corresponding subscripted values: for example, θ is the vector $(\theta_1, \theta_2, \dots)$, and π is the matrix $(\pi_{jj'})$. For variables that represent counts, I will use a dot in the subscript to represent a sum over corresponding individual counts; for example, $n_{j\cdot}$ is used to represent $\sum_{j'} n_{jj'}$, and $n_{\cdot\cdot}$ means $\sum_j \sum_{j'} n_{jj'}$.

3.1. Transition Dynamics in the HDP-HMM

The conventional HDP-HMM (Teh et al., 2006) is based on a Hierarchical Dirichlet Process which is defined as follows:

Each of a countably infinite set of states, indexed by j, receives a parameter vector $\theta_j \in \Theta$, according to base measure H. A top-level weight distribution, β, is drawn from the Griffith-Engels-McClosky (GEM) stick-breaking process with parameter $\gamma > 0$, so that state j has overall weight β_j, and an emission distribution which is parameterized by θ_j.

$$\theta_j \overset{\text{i.i.d.}}{\sim} H \tag{3.1}$$

$$\beta \sim \mathsf{GEM}(\gamma) \tag{3.2}$$

The actual transition distribution from state j, denoted by π_j is then drawn from a Dirichlet Process with concentration parameter α and base measure β:

$$\pi_j \overset{\text{i.i.d.}}{\sim} \mathsf{DP}(\alpha\beta) \qquad j = 1, 2, \dots \tag{3.3}$$

The hidden state sequence is then generated according to the π_j. Let z_t be the index of the chain's state at time t. Then we have

$$z_t \,|\, z_{t-1}, \pi_{z_{t-1}} \sim \pi_{z_{t-1}} \qquad t = 1, 2, \dots, T \tag{3.4}$$

where T is the length of the data sequence.

Finally, the emission distribution for state j is a function of θ_j, so that we have

$$y_t \mid z_t, \theta_{z_t} \sim F(\theta_{z_t}) \tag{3.5}$$

A shortcoming of this model is that the generative process does not take into account the fact that the set of source states is the same as the set of destination states: that is, that the distribution π_j has an element which corresponds to state j. Put another way, there is no special treatment of the diagonal of the transition matrix, so that self-transitions are no more likely *a priori* than transitions to any other state.

The Sticky HDP-HMM (Fox et al., 2008) addresses this issue by adding an extra mass of κ at location j to the base measure of the DP that generates π_j. That is, they replace (3.3) with

$$\pi_j \overset{\text{i.i.d.}}{\sim} \mathsf{DP}(\alpha\beta + \kappa\delta_j). \tag{3.6}$$

An alternative model is presented by Johnson and Willsky (2013), wherein state duration distributions are modeled separately, and ordinary self-transitions are ruled out. In both of these models, auxiliary latent variables are introduced to simplify conditional posterior distributions and facilitate Gibbs sampling. However, while both of these models have the useful property that self-transitions are treated as "special", they contain no notion of similarity for pairs of states that are not identical: in both cases, when $j \neq j'$, the prior probability of transitioning from j to j' depends only on the top-level stick weight associated with state j', and not on the identity or parameters of the previous state j.

3.2. An HDP-HMM With Local Transitions

The goal of the proposed model is to add to the transition model the concept of a transition to a "nearby" state, where nearness of j and j' may be defined deterministi-

cally or stochastically in terms of the emission parameters, θ_j and $\theta_{j'}$, or may be based on a latent state geometry that is a priori independent of the emission distribution.

In order to accomplish this, we first consider an alternative construction of the transition distributions, based on the Normalized Gamma Process representation of the Dirichlet Process (Ferguson, 1973).

3.2.1. A normalized Gamma Process representation of the HDP-HMM

Define a random measure, $\mu = \sum_{j=1}^{\infty} \pi_j \delta_{\theta_j}$, where for some choices of ω_j such that $\sum_{j=1}^{\infty} \omega_j < \infty$,

$$\pi_j \overset{\text{ind.}}{\sim} \mathsf{Gamma}(\omega_j, 1) \tag{3.7}$$

$$T = \sum_{j=1}^{\infty} \pi_j \tag{3.8}$$

$$\tilde{\pi}_j = \frac{\pi_j}{T} \tag{3.9}$$

$$\theta_j \overset{\text{i.i.d.}}{\sim} H \tag{3.10}$$

The summability constraint on the ω_j ensures that $T < \infty$ almost surely, since

$$T \sim \mathsf{Gamma}(\sum_j \omega_j, 1).$$

As shown by Paisley et al. (2011), for fixed $\{\omega_j\}$ and $\{\theta_j\}$, μ is distributed as a Dirichlet Process with base measure $\nu = \sum_{j=1}^{\infty} \omega_j \delta_{\theta_j}$. If we draw β from the GEM stick-breaking process and then draw a series $\{\mu_m\}_{m=1}^{M}$ of i.i.d. random measures from the above process, setting $\omega_j = \alpha\beta_j$ for some $\alpha > 0$, then this defines a Hierarchical Dirichlet Process. If, moreover, there is one μ_m associated with every state j, then we obtain the HDP-HMM.

We can thus write

$$\beta \sim \mathsf{GEM}(\gamma) \tag{3.11}$$

$$\theta_j \overset{\text{i.i.d.}}{\sim} H \tag{3.12}$$

$$\pi_{jj'} \overset{\text{ind.}}{\sim} \mathsf{Gamma}(\alpha\beta_{j'}, 1) \tag{3.13}$$

$$T_j = \sum_{j'=1}^{\infty} \pi_{jj'} \tag{3.14}$$

$$\tilde{\pi}_{jj'} = \frac{\pi_{jj'}}{T_j}, \tag{3.15}$$

where γ and α are prior concentration hyperparameters for the two DP levels,

$$p(z_t \,|\, z_{t-1}, \pi) = \tilde{\pi}_{z_{t-1} z_t} \tag{3.16}$$

and the observed data $\{y_t\}_{t \geq 1}$ is distributed as

$$y_t \,|\, z_t \overset{\text{ind.}}{\sim} F(\theta_{z_t}) \tag{3.17}$$

for some family, F of probability measures indexed by values of θ.

3.2.2. Promoting "local" transitions

In the preceding formulation, the transition distributions $\{\pi_j\}_{j=1}^{\infty}$ are independent conditioned on the top-level weights, β. Our goal is to relax this assumption, in order to allow for the possibility that there may be correlations between these distributions. We achieve this by introducing a notion of a geometry on the state space; that is, that some pairs of states are "near" to each other, and will thus tend to have similar transition probabilities associated with them.

We associate with each state a location, $\ell_j \in \mathcal{L}$, and define a "similarity function", $\phi : \mathcal{L} \times \mathcal{L} \to (0, 1]$, which returns for any pair of locations a measure of how "close" they are, where 1 corresponds to maximum similarity and 0 to maximum dissimilarity.

In order to remain agnostic about the extent to which ϕ is based on the emission parameters, I will use θ_j to represent the emission parameters for state j, and assume

that ℓ_j determines θ_j, but that ϕ may be based on any part of ℓ_j, which may include some, all, or none of the information contained in θ_j.

I will also use the shorthand $\phi_{jj'}$ to represent $\phi(\ell_j, \ell_{j'})$, for the sake of readability; but the reader should keep in mind that whenever ϕ appears in a conditional distribution, it is a constant if and only if ℓ is being conditioned on.

We can generalize the generative process defined in (3.11)-(3.15) as follows:

$$\beta \sim \mathsf{GEM}(\gamma) \tag{3.18}$$

$$(\theta_j, \ell_j) \overset{\text{i.i.d.}}{\sim} H \tag{3.19}$$

$$\pi_{jj'} \,|\, \beta, \ell \sim \mathsf{Gamma}(\alpha\beta_{j'}, \phi_{jj'}^{-1}) \tag{3.20}$$

$$T_j = \sum_{j'=1}^{\infty} \pi_{jj'} \tag{3.21}$$

$$\tilde{\pi}_{jj'} = \frac{\pi_{jj'}}{T_j} \tag{3.22}$$

$$y_t \,|\, z_t, \theta \overset{\text{ind.}}{\sim} F(\theta_{z_t}) \tag{3.23}$$

where H is now a measure on $\mathcal{L}$. In this new formuation, the expected value of $\pi_{jj'}$ is $\alpha\beta_{j'}\phi_{jj'}$. Since a similarity between one object and another should not exceed the similarity between an object and itself, we will assume that $\phi_{jj'} = 1$ if $j = j'$. We will also assume that the similarity function is symmetric, so that $\phi_{jj'} = \phi_{j'j}$ for all j, j'. As we will see, either or both of these assumptions could be relaxed if desired in a particular application, but derivations presented here will make both.

The above model is equivalent to simply drawing the $\pi_{jj'}$ as in (3.11) and scaling each one by $\phi_{jj'}$ prior to normalization, since it is a general property of Gamma distributions that, if $X \sim \mathsf{Gamma}(a, b)$, then $cX \sim \mathsf{Gamma}(a, b/c)$.

Unfortunately, this formulation complicates inference significantly, compared to the ordinary HDP-HMM, as the introduction of non-constant rate parameters to the prior on π destroys the conjugacy between π and z, and worse, the conditional likelihood function for π contains a sum which renders all entries within a row mutually dependent *a posteriori*.

3.3. The HDP-HMM-LT as a continuous-time Markov Jump Process with "failed" jumps

We can gain stronger intuition, as well as simplify posterior inference, by re-casting the HDP-HMM-LT described in the last section as a continuous time Markov Jump Process where some of the attempts to jump from one state to another fail, and where the failure probability increases as a function of the "distance" between the states.

Let ϕ be defined as in the last section, and let β, θ and π be defined as in the Normalized Gamma Process representation of the ordinary HDP-HMM. Specifically,

$$\beta \sim \mathsf{GEM}(\gamma) \tag{3.24}$$

$$(\theta_j, ell_j) \overset{\text{i.i.d.}}{\sim} H \tag{3.25}$$

$$\pi_{jj'} \,|\, \beta \sim \mathsf{Gamma}(\alpha\beta_{j'}, 1) \tag{3.26}$$

Now suppose that when the process is in state j, jumps to state j' are made at rate $\pi_{jj'}$. This defines a continuous-time Markov Process where the off-diagonal elements of the transition rate matrix are the off diagonal elements of the π matrix. In addition, self-jumps are allowed, and occur with rate π_{jj}.

If we only observe the jumps and not the durations between jumps, this is an ordinary Markov chain, whose transition matrix is obtained by appropriately normalizing the rows of π. If we do not observe the jumps themselves, but instead an observation is generated once per jump from a distribution that depends on the state being jumped to, then we have an ordinary HMM.

We modify this process as follows. Suppose that each jump attempt from state j to state j' has a chance of failing, which is an increasing function of the "distance" between the states. In particular, let the success probability be $\phi_{jj'}$ (recall that we assumed above that $0 \leq \phi_{jj'} \leq 1$ for all j, j'). Then, the rate of successful jumps from j to j' is $\pi_{jj'}\phi_{jj'}$, and the corresponding rate of unsuccessful jump attempts is $\pi_{jj'}(1-\phi_{jj'})$. To see this, denote by $N_{jj'}$ the total number of jump attempts to j' in a

unit interval of time spent in state j. Since we are assuming the process is Markovian, the total number of attempts is $\mathsf{Poisson}(\pi_{jj'})$ distributed. Conditioned on $N_{jj'}$, $n_{jj'}$ will be successful, where

$$n_{jj'} \mid N_{jj'} \sim \mathsf{Binom}(N_{jj'}, \phi_{jj'}) \tag{3.27}$$

It is easy to show (and well known) that the marginal distribution of $n_{jj'}$ is $\mathsf{Poisson}(\pi_{jj'}\phi_{jj'})$, and the marginal distribution of $N_{jj'} - n_{jj'}$ is $\mathsf{Poisson}(\pi_{jj'}(1-\phi_{jj'}))$. The rate of successful jumps from state j overall is then $T_j := \sum_{j'} \pi_{jj'}\phi_{jj'}$.

Let t index jumps, so that z_t indicates the tth state visited by the process (counting self-jumps as marking a transition to a new time step). Given that the process is in state j at discretized time t (that is, $z_t = j$), it is a standard property of Markov Processes that the probability that the destination z_{t+1} of the first successful jump is independent of the time since the last jump, and the probability that $z_{t+1} = j'$ is proportional to the rate, $\pi_{jj'}\phi_{jj'}$.

Let τ_t indicate the time elapsed between the $t-1$th and and tth successful jump (where we assume that the first observation occurs when the first successful jump from a "dummy" initial state is made). We have

$$\tau_t \mid z_{t-1} \sim \mathsf{Exponential}(T_{z_{t-1}}) \tag{3.28}$$

where τ_t is independent of z_t.

During this period, there will be some number of unsuccessful attempts to jump to each state, where the rate of unsuccessful jump attempts from state j to state j' is given by $\pi_{z_{t-1}}(1-\phi_{z_{t-1}j'})$. Denote by $\tilde{q}_{j't}$ the number of unsuccessful jump attempts to j' during the interval with duration τ_t. Then we have

$$\tilde{q}_{j't} \mid z_{t-1}, \tau_t \sim \mathsf{Poisson}(\tau_t \pi_{z_{t-1}j'}(1-\phi_{z_{t-1}j'})) \tag{3.29}$$

Define the following additional variables

$$\mathcal{T}_j = \{t \mid z_{t-1} = j\} \tag{3.30}$$

$$q_{jj'} = \sum_{t \in \mathcal{T}_j} \tilde{q}_{j't} \tag{3.31}$$

$$u_j = \sum_{t \in \mathcal{T}_j} \tau_t. \tag{3.32}$$

In addition, let $Q = (q_{jj'})_{j,j' \geq 1}$ be the matrix of unsuccessful jump attempt counts, and $u = (u_j)_{j \geq 1}$ be the vector whose jth entry is the total time spent in state j.

Since each of the τ_t with $t \in \mathcal{T}_j$ are i.i.d. $\mathsf{Exponential}(T_j)$, and since the sum of n Exponential random variables with shared scale λ has a $\mathsf{Gamma}(n, \lambda)$ distribution, we have

$$u_j \mid z, \pi, \ell \overset{\text{ind.}}{\sim} \mathsf{Gamma}(n_{j\cdot}, T_j) \tag{3.33}$$

where we define $n_{j\cdot} = \sum_{j'} n_{jj'}$, where $n_{jj'}$ is the number of successful jumps from state j to j' and $n_{j\cdot}$ is the total number of times that state j is visited.

Moreover, since the $\tilde{q}_{j't}$ with $t \in \mathcal{T}_j$ are Poisson distributed, the total number of failed attempts in the total duration u_j is

$$q_{jj'} \overset{\text{ind.}}{\sim} \mathsf{Poisson}(u_j \pi_{jj'} (1 - \phi_{jj'})). \tag{3.34}$$

Thus if we marginalize out the individual τ_t and $\tilde{q}_{j't}$, we have a joint distribution over z, u, and Q, conditioned on the transition rate matrix π and the success

probability matrix ϕ, which is

$$p(z, u, Q \mid \pi, \ell) = \left(\prod_{t=1}^{T} p(z_t \mid z_{t-1})\right) \prod_j p(u_j \mid z, \pi, \ell) \prod_{j'} p(q_{jj'} \mid u_j \pi_{jj'}, \phi_{jj'}) \tag{3.35}$$

$$= \left(\prod_t \frac{\pi_{z_{t-1}z_t}\phi_{z_{t-1}z_t}}{T_{z_{t-1}}}\right) \prod_j \frac{T_j^{n_{j\cdot}}}{\Gamma(n_{j\cdot})} u_j^{n_{j\cdot}-1} e^{-T_j u_j} \tag{3.36}$$

$$\times \prod_{j'} e^{-u_j \pi_{jj'}(1-\phi_{jj'})} u_j^{q_{jj'}} \pi_{jj'}^{q_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.37}$$

$$= \prod_j \Gamma(n_{j\cdot})^{-1} u_j^{n_{j\cdot}+q_{j\cdot}-1} \tag{3.38}$$

$$\times \prod_{j'} \pi_{jj'}^{n_{jj'}+q_{jj'}} \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} e^{-\pi_{jj'}\phi_{jj'}u_j} e^{-\pi_{jj'}(1-\phi_{jj'})u_j} (q_{jj'}!)^{-1} \tag{3.39}$$

$$= \prod_j \Gamma(n_{j\cdot})^{-1} u_j^{n_{j\cdot}+q_{j\cdot}-1} \prod_{j'} \pi_{jj'}^{n_{jj'}+q_{jj'}} \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} e^{-\pi_{jj'}u_j} (q_{jj'}!)^{-1} \tag{3.40}$$

3.3.1. An HDP-HSMM-LT modification

In any Hidden Markov Model, the distribution of the number of (discrete) time steps for which a given hidden state persists is by definition a Geometric distribution, where the "failure" parameter is the relevant entry on the diagonal of the transition matrix. The HDP-HMM and the HDP-HMM-LT as defined above are no exception. Although the Sticky HDP-HMM Fox et al. (2008) and the LT generalization presented here provide mechanisms for which the diagonal entries of the transition matrix will tend to have greater mass than the off-diagonal entries, they do not alter the Markovian assumption, which implies Geometric durations.

The HDP Hidden Semi-Markov Model (HDP-HSMM; Johnson and Willsky (2013)) gets around this restriction directly, by treating self-transitions as fundamentally distinct from all other transitions, and modeling state persistence durations directly. Should it be desireable to combine this property with the general similarity bias of the HDP-HMM-LT, it is trivial to modify the LT model to incorporate a separate

duration model. We can simply fix the diagonal elements of π to be zero, and allow D_t observations to be emitted i.i.d. from $F(\theta_{z_t})$ at jump t, where

$$D_t \mid z \overset{\text{ind.}}{\sim} g(\omega_{z_t}) \qquad \omega_j \overset{\text{i.i.d.}}{\sim} G \tag{3.41}$$

The likelihood then includes the additional term for the D_t, and the only inference step which is affected is that, instead of sampling z alone, we sample z and the D_t jointly, by defining

$$z_s^* = z_{\max\{T : s \leq \sum_{t=1}^{T} D_t\}} \tag{3.42}$$

where s ranges over the total number of observations, and associating a y_s observation sequence with each z_s^*. Inferences about ϕ are not affected, since the diagonal elements are assumed to be 1 anyway.

In the HDP-HSMM as it is presented in Johnson and Willsky (2013), zeroing out the diagonal of the transition matrix to isolate self-transitions to the separate duration model necessitates renormalization of the other entries so that the rows of the transition matrix are probability distributions. As a result, the conditional posterior for the matrix is no longer in an exponential family. To deal with this, Johnson and Willsky introduce auxiliary variables which can be interpreted as the diagonal entries of the transition matrix prior to zeroing out, as well as the number of self-transitions that would have occurred for each state had the transition matrix diagonal governed self-transitions instead of the separate duration model. Conditioned on these auxiliary variables, conjugacy between the transition matrix prior and likelihood is restored, and Gibbs sampling is able to proceed with exponential family updates.

In the LT model, on the other hand, we are already representing the transition matrix in unnormalized form, having rendered the entries conditionally independent given the z state sequence and the u holding times, so we are free to clamp the diagonal entries at zero without introducing a need for additional auxiliary variables in order to achieve semi-Markov dynamics.

3.3.2. Choice of similarity function

The similarity function, ϕ bears resemblance to a kernel function, as used in a number of machine learning methods, for example, to project data into a new feature space in for a Support Vector Machine, or as a covariance function for a Gaussian Process. Although the role of the similarity function here is similar, the restrictions on the similarity function here are different than the restrictions that define a valid kernel, which is why I have avoided using the term "kernel" in this context. For the HaMMLeT model, the similarity function must satisfy three properties:

1. It is bounded above.
2. It is nonnegative.
3. It is not identically zero.

Given the motivation for the model and the intuitive properties of a "similarity" function, it seems natural to assume additionally that ϕ is symmetric, and that it attains its upper bound when evaluating the similarity between a state and itself; that is, that $\phi_{jj'} = \phi_{j'j} \leq \phi_{jj} = 1$ for all j, j'. However, strictly speaking there is no technical requirement that this be the case; as long as the three conditions above are satisfied, the normalization constant in the Normalized Gamma Process representation will be finite and positive, and the Markov Process with Failed Jumps interpretation will hold (possibly after a global rescaling and absorbing the constant into the normalization so that $\phi_{jj'} \leq 1$ for all j, j').

Some natural choices of similarity function are the following, all of which depend on a distance function, d_λ.

1. The Laplacian kernel: $\phi(x, y) = \exp(-d_\lambda(x, y))$
2. The Gaussian kernel: $\phi(x, y) = \exp(-d_\lambda(x, y)^2)$

3. Functions of the form:

$$\phi(x, y) = (c + d_\lambda(x, y)^p)^{-q} \text{ for positive constants } c, p \text{ and } q$$

where 3 encompasses traditional kernel functions including the Cauchy, Generalized Student's t, rational quadratic, and inverse multiquadratic as special cases.

Notably absent from this list are kernel functions that take negative values, such as the linear, polynomial, log kernels, and wave kernels, among others.

3.3.3. Summary

I have defined the following augmented generative model for the HDP-H(S)MM-LT:

$$\beta \sim \mathsf{GEM}(\gamma) \tag{3.43}$$

$$(\theta_j, \ell_j) \overset{\text{i.i.d.}}{\sim} H \tag{3.44}$$

$$\pi_{jj'} \,|\, \beta \sim \mathsf{Gamma}(\alpha\beta_{j'}, 1) \tag{3.45}$$

$$z_t \,|\, z_{t-1}, \pi, \ell \sim \sum_j \left(\frac{\pi_{z_{t-1}j}\phi_{z_{t-1}j}}{\sum_{j'} \pi_{z_{t-1}j'}\phi_{z_{t-1}j'}} \right) \delta_j \tag{3.46}$$

$$u_j \,|\, z, \pi, \ell \overset{\text{ind.}}{\sim} \mathsf{Gamma}(n_{j\cdot}, \sum_{j'} \pi_{jj'}\phi_{jj'}) \tag{3.47}$$

$$q_{jj'} \,|\, u, \pi, \ell \overset{\text{ind.}}{\sim} \mathsf{Poisson}(u_j(1 - \phi_{jj'})\pi_{jj'}) \tag{3.48}$$

$$y_t \,|\, z, \theta \sim F(\theta_{z_t}) \tag{3.49}$$

If we are using the HSMM variant, then we simply fix π_{jj} to 0 for each j, draw

$$\omega_j \overset{\text{i.i.d.}}{\sim} G \tag{3.50}$$

$$D_t \,|\, z \overset{\text{ind.}}{\sim} g(\omega_{z_t}), \tag{3.51}$$

for chosen G and g, set

$$z^*_s = z_{\max\{T \,|\, s \le \sum_{t=1}^T D_t\}} \tag{3.52}$$

and replace (3.49) with

$$\mathbf{y}_s \,|\, z, \theta \sim F(\theta_{z^*_s}) \tag{3.53}$$

3.4. MCMC Inference in the "Failed Jumps" Representation

I develop a Gibbs sampling algorithm based on the Markov Process with Failed Jumps representation, augmenting the data with the duration variables u, the failed jump attempt count matrix, Q, as well as additional auxiliary variables which we will define below. In this representation the transition matrix is not modeled directly, but is a function of the unscaled transition matrix π and the similarity matrix ϕ. The full set of variables is partitioned into three blocks: (1) $\{\gamma, \alpha, \beta, \pi\}$, (2) $\{z, u, Q, \xi\}$, and (3) $\{\ell\}$, where ξ is a placeholder for an additional set of auxiliary variables that will be introduced below. The variables in each block are sampled jointly conditioned on the other two blocks. In some of the applications described in later chapters, blocks (2) and (3) can be sampled jointly, and if desired, the parameters of the ϕ function can be given priors and sampled as a separate block.

Since we are representing the transition matrix of the Markov chain explicitly, we approximate the stick-breaking process that produces β using a finite Dirichlet distribution with a number of components larger than we expect to need, forcing the remaining components to have zero weight.

Let J indicate the maximum number of states. Then, we approximate (3.24) with

$$\beta \,|\, \gamma \sim \text{Dirichlet}(\gamma/J, \ldots, \gamma/J) \tag{3.54}$$

This distribution converges weakly to the Stick-Breaking Process as $J \to \infty$ (Ishwaran and Zarepour, 2000). In practice, J is large enough when the vast majority of the probability mass in β is allocated to a strict subset of components, or when the latent state sequence z never uses all J available states, indicating that the data is well described by a number of states less than J.

3.4.1. Sampling π, β, α and γ

The joint conditional over γ, α, β and π given z, u, Q, ξ and θ will factor as

$$p(\gamma, \alpha, \beta, \pi \,|\, z, u, Q, \xi, \theta) = p(\gamma \,|\, \xi) p(\alpha \,|\, \xi) p(\beta \,|\, \gamma, \xi) p(\pi \,|\, \alpha, \beta, \theta, z) \tag{3.55}$$

I will derive these four factors in reverse order.

Sampling π The entries in π are conditionally independent given α and β, so we have the prior

$$p(\pi \,|\, \beta, \alpha) = \prod_j \prod_{j'} \Gamma(\alpha\beta_{j'})^{-1} \pi_{jj'}^{\alpha\beta_{j'}-1} \exp(-\pi_{jj'}), \tag{3.56}$$

and the likelihood given augmented data $\{z, u, Q\}$ given by (3.40). Combining these, we have

$$p(\pi, z, u, Q \,|\, \beta, \alpha, \theta) = \prod_j u_j^{n_{j\cdot}+q_{j\cdot}-1} \prod_{j'} \Gamma(\alpha\beta_{j'})^{-1} \pi_{jj'}^{\alpha\beta_{j'}+n_{jj'}+q_{jj'}-1} \tag{3.57}$$

$$\times\, e^{-(1+u_j)\pi_{jj'}} \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.58}$$

Conditioning on everything except π, we get

$$p(\pi \,|\, Q, u, z, \beta, \alpha, \theta) \propto \prod_j \prod_{j'} \pi_{jj'}^{\alpha\beta_{j'}+n_{jj'}+q_{jj'}-1} \exp(-(1+u_j)\pi_{jj'}) \tag{3.59}$$

and thus we see that the $\pi_{jj'}$ are conditionally independent given u, z and Q, and distributed according to

$$\pi_{jj'} \,|\, n_{jj'}, q_{jj'}, \beta_{j'}, \alpha \overset{\text{ind.}}{\sim} \mathsf{Gamma}(\alpha\beta_{j'} + n_{jj'} + q_{jj'}, 1 + u_j) \tag{3.60}$$

Sampling β Consider the conditional distribution of β having integrated out π. The prior density of β from (3.54) is

$$p(\beta \,|\, \gamma) = \frac{\Gamma(\gamma)}{\Gamma(\frac{\gamma}{J})^J} \prod_j \beta_j^{\frac{\gamma}{J}-1} \tag{3.61}$$

After integrating out π in (3.57), taking advantage of the fact that the terms containing π are proportional to the density of a Gamma distribution, we have

$$p(z, u, Q \mid \beta, \alpha, \gamma, \theta) = \prod_{j=1}^{J} u_j^{-1} \prod_{j'=1}^{J} u_j^{n_{jj'}+q_{jj'}} (1+u_j)^{-(\alpha\beta_{j'}+n_{jj'}+q_{jj'})} \tag{3.62}$$

$$\times \frac{\Gamma(\alpha\beta_{j'} + n_{jj'} + q_{jj'})}{\Gamma(\alpha\beta_{j'})} \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.63}$$

$$= \prod_{j=1}^{J} \Gamma(n_{j\cdot})^{-1} u_j^{-1} (1+u_j)^{-\alpha} \left(\frac{u_j}{1+u_j}\right)^{n_{j\cdot}+q_{j\cdot}} \tag{3.64}$$

$$\times \prod_{j'=1}^{J} \frac{\Gamma(\alpha\beta_{j'} + n_{jj'} + q_{jj'})}{\Gamma(\alpha\beta_{j'})} \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.65}$$

where we have used the fact that the β_j sum to 1. Therefore

$$p(\beta \mid z, u, Q, \alpha, \gamma, \theta) \propto \prod_{j=1}^{J} \beta_j^{\frac{\gamma}{J}-1} \prod_{j'=1}^{J} \frac{\Gamma(\alpha\beta_{j'} + n_{jj'} + q_{jj'})}{\Gamma(\alpha\beta_{j'})}. \tag{3.66}$$

Following (Teh et al., 2006), we can write the ratios of Gamma functions as polynomials in β_j, as

$$p(\beta \mid z, u, Q, \alpha, \gamma, \theta) \propto \prod_{j=1}^{J} \beta_j^{\frac{\gamma}{J}-1} \prod_{j'=1}^{J} \sum_{m_{jj'}=1}^{n_{jj'}} s(n_{jj'} + q_{jj'}, m_{jj'}) (\alpha\beta_{j'})^{m_{jj'}} \tag{3.67}$$

where $s(m, n)$ is an unsigned Stirling number of the first kind, which is used to represent the number of permutations of n elements such that there are m distinct cycles.

This admits an augmented data representation, where we introduce a random matrix $M = (m_{jj'})_{1 \le j,j' \le J}$, whose entries are conditionally independent given β, Q and z, with

$$p(m_{jj'} = m \mid \beta_{j'}, \alpha, n_{jj'}, q_{jj'}) = \frac{s(n_{jj'} + q_{jj'}, m) \alpha^m \beta_{j'}^m}{\sum_{m'=0}^{n_{jj'}+q_{jj'}} s(n_{jj'} + q_{jj'}, m') \alpha^{m'} \beta_{j'}^{m'}} \tag{3.68}$$

for integer m ranging between 0 and $n_{jj'} + q_{jj'}$. Note that $s(n, 0) = 0$ if $n > 0$, $s(0, 0) = 1$, $s(0, m) = 0$ if $m > 0$, and we have the recurrence relation $s(n+1, m) =$

$ns(n,m) + s(n,m-1)$, and so we could compute each of these coefficients explicitly; however, it is typically simpler and more computationally efficient to sample from this distribution than it is to enumerate its probabilities

For each $m_{jj'}$ we simply draw $n_{jj'}$ assignments of customers to tables according to the Chinese Restaurant Process and set $m_{jj'}$ to be the number of distinct tables realized; that is, assign the first customer to a table, setting $m_{jj'}$ to 1, and then, after n customers are assigned, assign the $n+1$th customer to a new table with probability $\alpha\beta_{j'}/(n+\alpha\beta_{j'})$, in which case we increment $m_{jj'}$, and to an existing table with probability $n/(n+\alpha)$, in which case we do not increment $m_{jj'}$.

Then, we have joint distribution

$$p(\beta, M \mid z, u, Q, \alpha, \gamma, \ell) \propto \prod_{j=1}^{J} \beta_j^{\frac{\gamma}{J}-1} \prod_{j'=1}^{J} s(n_{jj'} + q_{jj'}, m_{jj'}) \alpha^{m_{jj'}} \beta_{j'}^{m_{jj'}} \tag{3.69}$$

which yields (3.67) when marginalized over M. Again discarding constants in β and regrouping yields

$$p(\beta \mid M, z, u, \theta, \alpha, \gamma) \propto \prod_{j'=1}^{J} \beta_{j'}^{\frac{\gamma}{J}+m_{\cdot j'}-1} \tag{3.70}$$

which is Dirichlet:

$$\beta \mid M, \gamma \sim \text{Dirichlet}(\frac{\gamma}{J} + m_{\cdot 1}, \ldots, \frac{\gamma}{J} + m_{\cdot J}) \tag{3.71}$$

Sampling α and γ Assume that α and γ have Gamma priors, with

$$p(\alpha) = \frac{b_\alpha^{a_\alpha}}{\Gamma(a_\alpha)} \alpha^{a_\alpha - 1} \exp(-b_\alpha \alpha) \tag{3.72}$$

$$p(\gamma) = \frac{b_\gamma^{a_\gamma}}{\Gamma(a_\gamma)} \gamma^{a_{\gamma-1}} \exp(-b_\gamma \gamma) \tag{3.73}$$

Having integrated out π, we have

$$p(\beta, z, u, Q, M \mid \alpha, \gamma, \ell) = \frac{\Gamma(\gamma)}{\Gamma(\frac{\gamma}{J})^J} \alpha^{m_{\cdot\cdot}} \prod_{j=1}^{J} \beta_j^{\frac{\gamma}{J}+m_{\cdot j}-1} \Gamma(n_{j\cdot})^{-1} u_j^{-1} (1+u_j)^{-\alpha} \left(\frac{u_j}{1+u_j}\right)^{n_{j\cdot}+q_{j\cdot}} \tag{3.74}$$

$$\times \prod_{j'=1}^{J} s(n_{jj'}+q_{jj'}, m_{jj'}) \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.75}$$

We can also integrate out β, to yield

$$p(z, u, Q, M \mid \alpha, \gamma, \ell) = \alpha^{m_{\cdot\cdot}} e^{-\sum_{j''} \log(1+u_{j''})\alpha} \frac{\Gamma(\gamma)}{\Gamma(\gamma+m_{\cdot\cdot})} \tag{3.76}$$

$$\times \prod_{j} \frac{\Gamma(\frac{\gamma}{J}+m_{\cdot j})}{\Gamma(\frac{\gamma}{J})\Gamma(n_{j\cdot})} u_j^{-1} \left(\frac{u_j}{1+u_j}\right)^{n_{j\cdot}+q_{j\cdot}} \tag{3.77}$$

$$\times \prod_{j'=1}^{J} s(n_{jj'}+q_{jj'}, m_{jj'}) \phi_{jj'}^{n_{jj'}} (1-\phi_{jj'})^{q_{jj'}} (q_{jj'}!)^{-1} \tag{3.78}$$

demonstrating that α and γ are independent given ϕ (which in turn depends on ℓ) and the augmented data (z, u, Q, M), with

$$p(\alpha \mid z, u, Q, M, \ell) \propto \alpha^{a_\alpha + m_{\cdot\cdot}} \exp(-(b_\alpha + \sum_j \log(1+u_j))\alpha) \tag{3.79}$$

and

$$p(\gamma \mid z, u, Q, M, \ell) \propto \gamma^{a_\gamma - 1} \exp(-b_\gamma \gamma) \frac{\Gamma(\gamma) \prod_{j=1}^{J} \Gamma(\frac{\gamma}{J}+m_{\cdot j})}{\Gamma(\frac{\gamma}{J})^J \Gamma(\gamma+m_{\cdot\cdot})} \tag{3.80}$$

So we see that

$$\alpha \mid z, u, Q, M, \ell \sim \mathsf{Gamma}(a_\alpha + m_{\cdot\cdot}, b_\alpha + \sum_j \log(1+u_j)) \tag{3.81}$$

To sample γ, we introduce a new set of auxiliary variables, $r = (r_1, \ldots, r_J)$ and t with the following distributions:

$$p(r_{j'} = r \mid m_{\cdot j'}, \gamma) = \frac{\Gamma(\frac{\gamma}{J})}{\Gamma(\frac{\gamma}{J}+m_{\cdot j'})} s(m_{\cdot j'}, r) \left(\frac{\gamma}{J}\right)^r \qquad r = 1, \ldots, m_{\cdot j} \tag{3.82}$$

$$p(t \mid m_{\cdot\cdot}\gamma) = \frac{\Gamma(\gamma+m_{\cdot\cdot})}{\Gamma(\gamma)\Gamma(m_{\cdot\cdot})} t^{\gamma-1}(1-t)^{m_{\cdot\cdot}-1} \qquad t \in (0,1) \tag{3.83}$$

so that

$$p(\gamma, r, t \mid M) \propto \gamma^{a_\gamma - 1} \exp(-b_\gamma \gamma) t^{\gamma - 1} (1-t)^{m_{\cdot\cdot} - 1} \prod_{j'=1}^{J} s(m_{\cdot j'}, r_{j'}) \left(\frac{\gamma}{J}\right)^{r_{j'}} \tag{3.84}$$

and

$$p(\gamma \mid r, t) \propto \gamma^{a_\gamma + r_\cdot - 1} \exp(-(b_\gamma - \log(t))\gamma), \tag{3.85}$$

which is to say

$$\gamma \mid r, t, z, u, Q, M, \ell \sim \mathsf{Gamma}(a_\gamma + r_\cdot, b_\gamma - \log(t)) \tag{3.86}$$

Summary I have made the following additional assumptions about the generative model in this section:

$$\gamma \sim \mathsf{Gamma}(a_\gamma, b_\gamma) \qquad \alpha \sim \mathsf{Gamma}(a_\alpha, b_\alpha) \tag{3.87}$$

The joint conditional over γ, α, β and π given z, u, Q, M, r, t and ℓ factors as

$$p(\gamma, \alpha, \beta, \pi \mid z, u, Q, r, t, \ell) = p(\gamma \mid r, t) p(\alpha \mid u, M) p(\beta \mid \gamma, M) p(\pi \mid \alpha, \beta, z, u, Q) \tag{3.88}$$

where

$$\gamma \mid r, t \sim \mathsf{Gamma}(a_\gamma + r_\cdot, b_\gamma - \log(t)) \tag{3.89}$$

$$\alpha \mid u, M \sim \mathsf{Gamma}(a_\alpha + m_{\cdot\cdot}, b_\alpha + \sum_j \log(1 + u_j)) \tag{3.90}$$

$$\beta \mid \gamma, M \sim \text{Dirichlet}(\frac{\gamma}{J} + m_{\cdot 1}, \ldots, \frac{\gamma}{J} + m_{\cdot J}) \tag{3.91}$$

$$\pi_{jj'} \mid \alpha, \beta_{j'}, z, u, Q \overset{\text{ind.}}{\sim} \mathsf{Gamma}(\alpha\beta_{j'} + n_{jj'} + q_{jj'}, 1 + u_j) \tag{3.92}$$

3.4.2. Sampling z and the auxiliary variables

The hidden state sequence, z, is sampled jointly with the auxiliary variables, which consist of u, M, Q, r and t. The joint conditional distribution of these variables is

defined directly by the generative model:

$$p(z, u, Q, M, r, t \,|\, \pi, \beta, \alpha, \gamma, \ell) = p(z \,|\, \pi, \ell) p(u \,|\, z, \pi, \ell) p(Q \,|\, u, \pi, \ell) p(M \,|\, z, Q, \alpha, \beta) \tag{3.93}$$

$$\times\, p(r \,|\, \gamma, M) p(t \,|\, \gamma, M) \tag{3.94}$$

Since we are representing the transition matrix explicitly, we can sample the entire sequence z at once with the forward-backward algorithm, as in an ordinary HMM (or, if we are employing the HSMM variant described in Sec. 3.3.1, then we can use the modified message passing scheme for HSMMs described by Johnson and Willsky (2013)). Having done this, we can sample u, Q, M, r and t from their forward distributions. To summarize, we have

$$u_j \,|\, z, \pi, \ell \overset{\text{ind.}}{\sim} \mathsf{Gamma}(n_{j\cdot}, \sum_{j'} \pi_{jj'} \phi_{jj'}) \tag{3.95}$$

$$q_{jj'} \,|\, u_j, \pi_{jj'}, \phi_{jj'} \overset{\text{ind.}}{\sim} \mathsf{Poisson}(u_j (1 - \phi_{jj'}) \pi_{jj'}) \tag{3.96}$$

$$m_{jj'} \,|\, n_{jj'}, q_{jj'}, \beta_{j'}, \alpha \overset{\text{ind.}}{\sim} \frac{\Gamma(\alpha \beta_j)}{\Gamma(\alpha \beta_j + n_{jj'} + q_{jj'})} \sum_{m=1}^{n_{jj'} + q_{jj'}} s(n_{jj'} + q_{jj'}, m) \alpha^m \beta_{j'}^m \delta_m \tag{3.97}$$

$$r_j \,|\, m_{\cdot j}, \gamma \overset{\text{ind.}}{\sim} \frac{\Gamma(\frac{\gamma}{J})}{\Gamma(\frac{\gamma}{J} + m_{\cdot j})} \sum_{r=1}^{m_{j\cdot}} s(m_{\cdot j}, r) \left(\frac{\gamma}{J}\right)^r \delta_r \tag{3.98}$$

$$t \,|\, \gamma, M \sim \mathsf{Beta}(\gamma, m_{\cdot\cdot}) \tag{3.99}$$

3.4.3. Sampling state and emission parameters

The state locations ℓ influence the transition matrix, π and the auxiliary vector q through the similarity matrix ϕ, while the state parameters θ control the emission distributions. We have likelihood factors

$$p(z, Q \,|\, \ell, \theta) \propto \prod_j \prod_{j'} \phi_{jj'}^{n_{jj'}} (1 - \phi_{jj'})^{q_{jj'}} \tag{3.100}$$

$$p(y \,|\, z, \theta) = \prod_{t=1}^{T} f(y_t; \theta_{z_t}) \tag{3.101}$$

where, recall, $n_{jj'}$ counts the number of times that the state sequence transitions from state j to state j', and proportionality is with respect to variation in ℓ and θ.

The parameter space for the hidden states, the associated prior H on each (ℓ_j, θ_j) pair, and the similarity function ϕ, are application-specific, thus I now turn to two individual applications with qualitatively different state spaces and emission distributions, derive inference methods for each, and present experiments on synthetic and real data.

3.5. Use Cases

In Chapter 4, I describe an application of the HaMMLeT model to a synthetic dataset designed to mimic a speaker diarization or blind source separation task. Here, each latent state corresponds to a description of which of several sound sources is active at a given time step, and where the observed data is a set of signals picked up from several microphones distributed around the room, each of which picks up all of the sound sources with varying sensitivities, and the goal is to determine who is speaking when. In this application the latent states are represented as binary vectors, similarity is based on Hamming distance between binary vectors, and the emission model is a multivariate linear-Gaussian model.

In Chapter 5 I describe an application to unsupervised learning of musical structure. The data is a sequence of chords in a musical composition, represented as a single symbol, and the goal is to infer a set of chord equivalence classes and a transition model between equivalence classes. Unlike in the previous application, similarity between states is modeled separately from the emission distributions, so that ϕ is based on the ℓ part of the state parameters, and θ is a disjoint set of parameters governing the emission distribution.

Finally, in Chapter 6 I describe a potential future application to power disaggregation, in which the observation sequence consists the amount of power used by a house

at various times throughout a day, and the goal is to separate the aggregated power signal into several signals corresponding to a set of appliances (oven, air conditioning, lighting, etc.). Unlike the cocktail party setting, each latent signal may have more than two levels, and so in addition to inferring what appliances are on when, the model needs to discover how much power each appliance uses in each of its latent states. Here, latent states are vectors where each entry is a categorical label, the number of categories per channel is unknown, the similarity model is again based on Hamming distance between binary vectors, and the emission model is linear-Gaussian.

4. BINARY VECTOR STATES: SPEAKER DIARIZATION

4.1. Binary State Vectors

I consider here the case where both ℓ_j and θ_j refer to the same finite D-dimensional binary vector, and the emission distribution is based on a noisy linear transformation of the state vector according to a $D \times K$ weight matrix W, so that each K-dimensional observation is $\mathcal{N}(W^\mathsf{T}\theta_j, \Sigma)$, where $\mathbf{W}$ and Σ are additional hyperparameters in the model. For the experiments discussed in this chapter, I will assume that Σ does not depend on j, but this assumption is easily relaxed if appropriate. For finite-length binary vector states, the set of possible states is finite, and so on its face it may seem that a nonparametric model is unnecessary. However, if D is reasonably large, it is likely that most of the 2^D possible states are vanishingly unlikely (and, in fact, the number of observations may well be less than 2^D), and so we would like a model that encourages the selection of a sparse set of states. Moreover, there could be more than one state with the same emission parameters, but with different transition dynamics.

We require a similarity function, $\phi(\theta_j, \theta_{j'})$, which varies between 0 to 1, and is equal to 1 if and only if $\theta_j = \theta_{j'}$. A natural choice in this setting is the Laplacian kernel:

$$\phi_{jj'} = \phi(\theta_j, \theta_{j'}) = \exp(-\lambda \Delta_{jj'}) \tag{4.1}$$

where $\Delta_{jj'd} = |\theta_{jd} - \theta_{j'd}|$, $\Delta_{jj'} = \sum_{d=1}^{D} \Delta_{jj'}$ is the Hamming distance between θ_j and $\theta_{j'}$, and $\lambda \geq 0$ (if $\lambda = 0$, the $\phi_{jj'}$ are all 1, and so do not have any influence, reducing the model to an ordinary HDP-HMM).

Before describing individual experiments, I describe the additional inference steps needed for these variables.

4.1.1. Additional inference steps

Sampling θ We put independent Beta-Bernoulli priors on each coordinate of θ. Each coordinate θ_{jd} can be sampled conditioned on all the others and the coordinate-wise prior means, $\{\mu_d\}$, which we sample in turn conditioned on the θs. Since individual coordinates are binary, each one has a Bernoulli posterior, whose parameter we now derive by computing the posterior log-odds, that is we will derive

$$\log \frac{p(\theta_{jd} = 1 \,|\, \theta \setminus \theta_{jd}, z, Q, Y)}{p(\theta_{jd} = 0 \,|\, \theta \setminus \theta_{jd}, z, Q, Y)} \tag{4.2}$$

Since θ_{jd} affects both the similarity matrix and the emission distribution, the posterior log-odds has three components: the prior log odds, the likelihood due to the successful and failed transitions (specifically the outgoing transition counts from state j, $n_{jj'}$ and $q_{jj'}$, as well as the incoming transition counts to state j, $n_{j'j}$ and $q_{j'j}$, across all j'), and the likelihood due to the data, y, for those t such that $z_t = j$.

The prior log odds is simply $\log(\mu_d/(1-\mu_d))$. Next, we compute the likelihood component due to the transition attempts, that is

$$\log \frac{p(z, Q \,|\, \theta_{jd} = 1, \theta \setminus \theta_{jd}, \lambda)}{p(z, Q \,|\, \theta_{jd} = 0, \theta \setminus \theta_{jd}, \lambda)}$$

Let

$$\phi_{jj'-d} = \exp(-\lambda(\Delta_{jj'} - \Delta_{jj'd})), \tag{4.3}$$

that is, the similarity between states j and j' with the dth coordinate ignored, so that $\phi_{jj'} = \phi_{jj'-d} e^{-\lambda \Delta_{jj'd}}$.

Since the matrix ϕ is assumed to be symmetric, we have

$$\frac{p(z, Q \,|\, \theta_{jd} = 1, \theta \setminus \theta_{jd})}{p(z, Q \,|\, \theta_{jd} = 0, \theta \setminus \theta_{jd})} \propto \prod_{j' \neq j} \frac{e^{-\lambda(n_{jj'}+n_{j'j})\left|1-\theta_{j'd}\right|}\left(1 - \phi_{jj'-d} e^{-\lambda\left|1-\theta_{j'd}\right|}\right)^{q_{jj'}+q_{j'j}}}{e^{-\lambda(n_{jj'}+n_{j'j})\left|\theta_{j'd}\right|}\left(1 - \phi_{jj'-d} e^{-\lambda\left|\theta_{j'd}\right|}\right)^{q_{jj'}+q_{j'j}}} \tag{4.4}$$

$$= e^{-\lambda(c_{jd0} - c_{jd1})} \prod_{j' \neq j} \left(\frac{1 - \phi_{jj'-d} e^{-\lambda}}{1 - \phi_{jj'-d}}\right)^{(-1)^{\theta_{j'd}}(q_{jj'}+q_{j'j})} \tag{4.5}$$

where c_{jd0} and c_{jd1} are the number of successful jumps to or from state j, to or from states with a 0 or 1, respectively, in position d. That is,

$$c_{jd0} = \sum_{\{j' \,|\, \theta_{j'd}=0\}} n_{jj'} + n_{j'j} \qquad c_{jd1} = \sum_{\{j' \,|\, \theta_{j'd}=1\}} n_{jj'} + n_{j'j} \tag{4.6}$$

We assume the observed data Y consists of a $T \times K$ matrix, where the tth row $y_t = (y_{t1}, \ldots, y_{tK})$ is a K-dimensional feature vector associated with time t, and let W be a $D \times K$ weight matrix with kth column w_k that maps D dimensional binary state vectors to means in the K-dimensional observation space. Then the part of the likelihood function from time t is

$$p(y_t \,|\, z_t, \theta_{z_t}) = \mathcal{N}(y_t \,|\, W^{\mathsf{T}}\theta_{z_t}, \Sigma) \tag{4.7}$$

In the experiments discussed here, Σ is asssumed diagonal with entries σ_k^2, $k = 1, \ldots, K$, and does not depend on j. Under this assumption, we have

$$p(y_t \,|\, W^{\mathsf{T}}\theta_{z_t}) = \prod_{k=1}^{K} \mathcal{N}(y_t \,|\, w_k^{\mathsf{T}}\theta_{z_t}, \sigma_k^2). \tag{4.8}$$

The coordinate θ_{jd} affects the likelihood only through the mean, $W^{\mathsf{T}}\theta_j$. Expanding this mean to isolate the effect of θ_{jd}, we have

$$w_k^{\mathsf{T}}\theta_j = \sum_d w_{dk}\theta_{jd}$$

and so we see that setting $\theta_{jd} = 1$ vs 0 shifts the kth coordinate of the mean, $W_{\mathsf{T}}\theta_j$, by w_{dk}. Denote by θ^* the matrix whose tth row consists of the binary state vector θ_{z_t}, and let $X^* = W^{\mathsf{T}}\theta^*$. As shorthand, write $x_{tk}^{(-d)}$ to be the (t,k) entry in X^* assuming that $\theta_{jd} = 0$. Then the corresponding value if $\theta_{jd} = 1$ is $x_{tk}^{(-d)} + w_{dk}$. Since the y_t are conditionally independent given the state sequence, z, we can write the log likelihood ratio for $\theta_{jd} = 1$ versus $\theta_{jd} = 0$ as

$$\sum_{t:z_t=j} \sum_{k=1}^{K} \log\left(\frac{\mathcal{N}(y_{tk} \,|\, x_{tk}^{(-d)} + w_{dk}, \sigma_k^2))}{\mathcal{N}(y_{tk} \,|\, x_{tk}^{(-d)}, \sigma_k^2)}\right) = \sum_{t:z_t=j} \sum_{k=1}^{K} -\frac{w_{dk}}{\sigma_k^2}(y_{tk} - x_{tk}^{(-d)} + \frac{1}{2}w_{dk}) \tag{4.9}$$

All together, we have

$$\log\left(\frac{p(\theta_{jd}=1\,|\,Y,z,Q,\theta\setminus\theta_{jd})}{p(\theta_{jd}=0\,|\,Y,z,Q,\theta\setminus\theta_{jd})}\right) \tag{4.10}$$

$$=\log\left(\frac{p(\theta_{jd}=1)p(z,Q\,|\,\theta_{jd}=1,\theta\setminus\theta_{jd})p(Y\,|\,z,\theta_{jd}=1,\theta\setminus\theta_{jd})}{p(\theta_{jd}=0)p(z,Q\,|\,\theta_{jd}=0,\theta\setminus\theta_{jd})p(Y\,|\,z,\theta_{jd}=0,\theta\setminus\theta_{jd})}\right) \tag{4.11}$$

$$=\log\left(\frac{\mu_d}{1-\mu_d}\right) \tag{4.12}$$

$$+(c_{jd1}-c_{jd0})\lambda+\sum_{j'\neq j}(-1)^{\theta_{j'd}}(q_{jj'}+q_{j'j})\log\left(\frac{1-\phi_{jj'}^{(-d)}e^{-\lambda}}{1-\phi_{jj'}^{(-d)}}\right) \tag{4.13}$$

$$+\sum_{t:z_t=j}\sum_{k=1}^{K}-\frac{w_{dk}}{\sigma_k^2}(y_{tk}-x_{tk}^{(-d)}+\frac{1}{2}w_{dk}) \tag{4.14}$$

Sampling μ Sampling the μ_d is straightforward with a Beta prior. Suppose

$$\mu_d \overset{ind}{\sim} \mathsf{Beta}(a_\mu, b_\mu) \tag{4.15}$$

Then, conditioned on θ the μ_d are independent with

$$\mu_d\,|\,\theta \sim \mathsf{Beta}(a_\mu+\sum_j\theta_{jd}, b_\mu+\sum_j(1-\theta_{jd})) \tag{4.16}$$

Sampling W and Σ Conditioned on the state matrix θ and the data matrix Y, the weight matrix W can be sampled as well using standard methods for Bayesian linear regression. We place a zero mean Normal prior on each element of W (including a row of intercept terms), resulting in a multivariate Normal posterior for each column. For the experiments reported below, we constrain Σ to be a diagonal matrix, and place an Inverse Gamma prior on the variances, resulting in conjugate updates.

Define the prior:

$$p(W)=\prod_{k=1}^{K}\prod_{d=1}^{D}\mathcal{N}(w_{dk}\,|\,0,\sigma_0^2). \tag{4.17}$$

The likelihood is

$$p(Y\,|\,W,\theta)=\prod_{t=1}^{T}\prod_{k=1}^{K}p(y_{tk};x_{tk})=\prod_{t=1}^{T}\prod_{k=1}^{K}\mathcal{N}(y_{tk}\,|\,x_{tk},\sigma_k^2) \tag{4.18}$$

Then it is a standard result from Bayesian linear modeling that

$$p(W \mid \theta, Y) = \prod_{k=1}^{K} \mathcal{N}(\left(\sigma_0^{-2}\mathbf{I} + \Sigma^{-1}\theta^{*\mathsf{T}}\theta^*\right)^{-1}\Sigma^{-1}\theta^{*\mathsf{T}}Y, \left(\sigma_0^{-2}\mathbf{I} + \Sigma^{-1}\theta^{*\mathsf{T}}\theta^*\right)^{-1}) \quad (4.19)$$

Sampling λ The Laplacian kernel ϕ is defined as $\phi(\theta_j, \theta_{j'}) = e^{-\lambda d(\theta_j, \theta_{j'})}$, where in our case d is Hamming distance. The parameter λ governs the connection between θ and ϕ. Writing (3.100) in terms of λ and the distance matrix Δ gives the likelihood

$$p(z, Q \mid \lambda, \theta) \propto \prod_j \prod_{j'} e^{-\lambda \Delta_{jj'} n_{jj'}} (1 - e^{-\lambda \Delta_{jj'}})^{q_{jj'}} \quad (4.20)$$

We put an $\mathsf{Exponential}(b_\lambda)$ prior on λ, which yields a posterior density

$$p(\lambda \mid z, Q, \theta) \propto e^{-(b_\lambda + \sum_j \sum_{j'} \Delta_{jj'} n_{jj'})\lambda} \prod_j \prod_{j'} (1 - e^{-\lambda \Delta_{jj'}})^{q_{jj'}} \quad (4.21)$$

This density is log-concave provided that, for each j, $\Delta_{jj'} > 0$ for at least one j', since

$$-\frac{d^2 \log(p(\lambda \mid z, Q, \theta))}{d\lambda^2} = \sum_{\{(j,j') \mid \Delta_{jj'} > 0\}} \frac{\Delta_{jj'}^2 q_{jj'} e^{\lambda \Delta_{jj'}}}{(e^{\lambda \Delta_{jj'}} - 1)^2} > 0 \quad (4.22)$$

and so we can use Adaptive Rejection Sampling (Gilks and Wild, 1992) to sample from it. To do this we need to compute the log density and its first derivative, which are given by

$$h(\lambda) = -(b_\lambda + \sum_{\{(j,j') \mid \Delta_{jj'} > 0\}} \Delta_{jj'} n_{jj'})\lambda + \sum_{\{(j,j') \mid \Delta_{jj'} > 0\}} q_{jj'} \log(1 - e^{-\lambda \Delta_{jj'}}) \quad (4.23)$$

$$h'(\lambda) = -(b_\lambda + \sum_{\{(j,j') \mid \Delta_{jj'} > 0\}} \Delta_{jj'} n_{jj'}) + \sum_{\{(j,j') \mid \Delta_{jj'} > 0\}} \frac{q_{jj'} \Delta_{jj'}}{e^{\lambda \Delta_{jj'}} - 1} \quad (4.24)$$

4.1.2. Summary

I have made the following assumptions about the representation of the hidden states and observed data in this subsection: (1) States are distinguished via a D-dimensional binary parameter vector, to which both ℓ_j and θ_j refer, (2) the similarity ϕ between

states j and j' is the Laplacian kernel with respect to Hamming distance between the respective ℓ vectors, and has a decay parameter λ, and (3) Y consists of K real-valued features associated with each time step t. In addition, the following priors are applied:

$$\mu_d \overset{i.i.d}{\sim} \mathsf{Beta}(a_\mu, b_\mu) \tag{4.25}$$

$$\lambda \sim \mathsf{Exponential}(b_\lambda) \tag{4.26}$$

$$\theta_{jd} \,|\, \mu \overset{ind}{\sim} \mathsf{Bernoulli}\,(\mu_d) \tag{4.27}$$

$$w_{dk} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_0^2) \tag{4.28}$$

$$y_{tk} \,|\, W, z, \theta \overset{ind}{\sim} \mathcal{N}(w_k^T \theta_{z_t}, \Sigma) \tag{4.29}$$

I introduce Gibbs blocks corresponding to (1) each θ_{jd} individually, (2) the vector μ, (3) the decay parameter λ, and (4) the weight matrix W. We have conditional posterior distributions

$$\theta_{jd} \,|\, \theta \setminus \theta_{jd}, z, Q, \mu, \lambda, W, Y^* \sim \mathsf{Bernoulli}\left(\frac{e^{\zeta_{jd}}}{1 + e^{\zeta_{jd}}}\right) \tag{4.30}$$

$$\mu_d \,|\, \theta, \dots \overset{ind}{\sim} \mathsf{Beta}(a_\mu + \sum_j \theta_{jd}, b_\mu + \sum_j (1 - \theta_{jd})) \tag{4.31}$$

$$p(\lambda \,|\, z, Q, \theta, \dots) \propto e^{-(b_\lambda + \sum_j \sum_{j'} \Delta_{jj'} n_{jj'})\lambda} \prod_j \prod_{j'} (1 - e^{-\lambda \Delta_{jj'}})^{q_{jj'}} \tag{4.32}$$

$$W \,|\, \theta, Y, z, \dots \overset{ind}{\sim} \mathcal{N}(\left(\sigma_0^{-2}\mathbf{I} + \Sigma^{-1}\theta^{*\mathsf{T}}\theta^*\right)^{-1} \Sigma^{-1}\theta^{*\mathsf{T}} Y, \left(\sigma_0^{-2}\mathbf{I} + \Sigma^{-1}\theta^{*\mathsf{T}}\theta^*\right)^{-1}) \tag{4.33}$$

where $\Delta_{jj'} = \left|\left|\theta_j - \theta'_j\right|\right|_{L_1}$ and

$$\zeta_{jd} = \log\left(\frac{\mu_d}{1 - \mu_d}\right) + (c_{jd1} - c_{jd0})\lambda + \sum_{j' \neq j} (-1)^{\theta_{j'd}} (q_{jj'} + q_{j'j}) \log\left(\frac{1 - \phi_{jj'}^{(-d)} e^{-\lambda}}{1 - \phi_{jj'}^{(-d)}}\right)$$
$$- \sum_{\{t \,|\, z_t = j\}} \sum_{k=1}^{K} \frac{w_{dk}}{\sigma_k^2} (y_{tk}^* - x_{tk}^{(-d)} + \frac{1}{2} w_{dk}) \tag{4.34}$$

All distributions can be sampled from directly except for λ, which requires Adaptive

Rejection Sampling, with the equations

$$h(\lambda) = -(b_\lambda + \sum_{\{(j,j') \,|\, \Delta_{jj'} > 0\}} \Delta_{jj'} n_{jj'})\lambda + \sum_{\{(j,j') \,|\, \Delta_{jj'} > 0\}} q_{jj'} \log(1 - e^{-\lambda \Delta_{jj'}}) \quad (4.35)$$

$$h'(\lambda) = -(b_\lambda + \sum_{\{(j,j') \,|\, \Delta_{jj'} > 0\}} \Delta_{jj'} n_{jj'}) + \sum_{\{(j,j') \,|\, \Delta_{jj'} > 0\}} \frac{q_{jj'} \Delta_{jj'}}{e^{\lambda \Delta_{jj'}} - 1} \quad (4.36)$$

4.2. "Cocktail Party" Data

The binary state instantiation of the model was evaluated using a speaker diarization/blind source separation task known as the "cocktail party" problem. In such a task, the observable data consists of one or more audio signals, discretized into T segments, where the signal at time t is assumed to arise from multiple speakers, some subset of whom are speaking. The inference problem is to recover which speakers are speaking when. In this experiment it is assumed that speakers are grouped into conversations, and take turns speaking within conversation, a modification of a standard speaker diarization or "cocktail party" blind-source separation setting. An example binary state sequence with three groups of 3, 2 and 2 speakers is shown in Fig. 4.1, with time on the horizontal axis and speaker on the vertical axis, so that the full latent binary state at a given time is represented by an individual column.

In this task, there are naively 2^D possible states[1], where each state indicates which of D speakers is speaking at a particular time step. However, due to the conversational grouping, if at most one speaker in a conversation can be speaking at any given time (a natural assumption of within-group turn-taking), the state space is constrained, with only $\prod_c (s_c + 1)$ states possible (with perhaps a small probability of other combinations occurring due to occasional overlap), where s_c is the number of speakers in conversation c, and the extra state corresponds to the case where none of the s_c speakers is speaking.

[1]Strictly speaking, even though the number of distinct binary vectors is finite, there could be more than one state with the same binary vector but different transition probabilities.

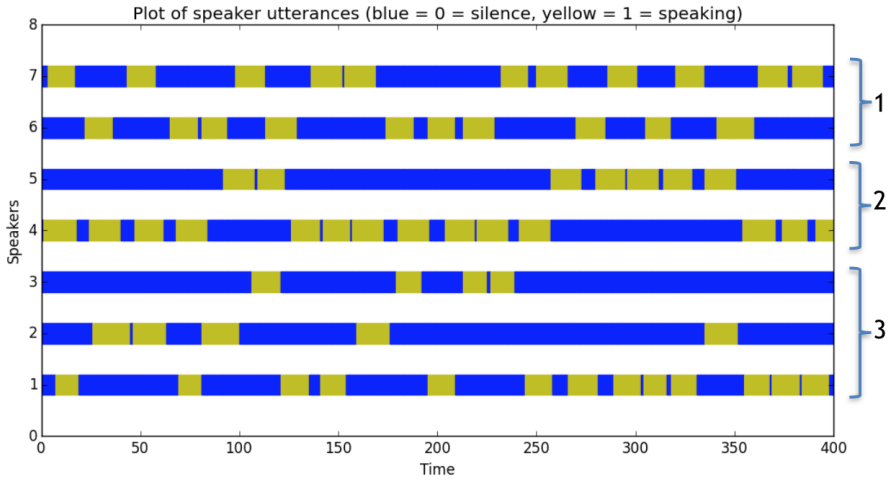


Figure 4.1: Example latent binary vector sequence for synthetic "cocktail party with conversational groups" data. Conversational groups are indicated by the brackets on the right.

Two datasets were used for evaluation, one synthetic and one using utterances from the the PASCAL 1st Speech Separation Challenge[2]. In both cases, the underlying signal consisted of 16 simultaneous speakers, with the signal matrix linearly mapped to 12 microphone channels. The 16 speakers were grouped into 4 conversational groups of 4 speakers each, where speakers within a conversation took turns speaking. Hence there are $2^{16} = 65536$ states possible assuming that any combination of speakers can be active at a given time; however, because of turn-taking only $5^4 = 625$ of these can actually occur.

Synthetic Data For the synthetic cocktail party data, the turn sequence within each conversation was generated using a Hidden Semi-Markov Model (HSMM) with Poisson distributed state-durations, in which the state sequence had s_c states, with pauses with shorter Poisson duration inserted between each "sentence". The states within conversations were then mapped to a binary vector with s_c entries, where silence is represented by all zeroes, and speaker s speaking corresponds to a 1 in

[2] `http://laslab.org/SpeechSeparationChallenge/`

position s. The binary vectors were concatenated across conversations to yield latent states consisting of length S binary vectors. To simulate speakers being recorded by K microphones, weights mapping speakers to microphones were generated independently from a $\mathsf{Unif}(0, 1)$ distribution, resulting in a $D \times K$ weight matrix, W. A constant "background sound level" parameter for each microphone was added as well, also $\mathsf{Unif}(0, 1)$, and independent $\mathcal{N}(0, \sigma_k^2)$ noise was added to each time step at microphone k.

Transition and emission parameters were generated from conjugate priors to the Poisson HSMM. The data set consisted of four conversations of four speakers each, and 12 microphones, so that $D = 16$, and $K = 12$. There are therefore $2^{16} = 65536$ possible binary vector-valued states, but only $(4+1)^4 = 625$, less than 1%, can actually occur. The noise variance σ_k^2 was set to a constant of $1/10$ for all $k = 1, \ldots, K$.

PASCAL Dataset The dataset for this experiment was constructed using a pool of 500 utterances from the PASCAL 1st Speech Separation Challenge[3]. In this data, each "turn" within a conversation consists of a single sentence (average duration $\sim$ 3s) and turn orders within a conversation were randomly generated, with random pauses distributed $\mathcal{N}(1/4s, (1/4s)^2)$ inserted between sentences. Every time a speaker has a turn, the sentence is drawn randomly from the 500 sentences uttered by that speaker in the data. The conversations continued for 40s, and the signal was down-sampled to length 2000. The 'on' portions of each speaker's signal were normalized to have amplitudes with mean 1 and standard deviation 0.5. An additional column of 1s was added to the speaker signal matrix, representing background noise. The resulting signal matrix, $\boldsymbol{\eta}$, was thus 2000×17 and the weight matrix was 17×12. Following Gael et al. (2009) and Valera et al. (2015), the weights were drawn independently from a $\mathsf{Unif}(0, 1)$ distribution, and independent $\mathcal{N}(0, 0.3^2)$ noise was added to each entry of the observation matrix.

[3] `http://laslab.org/SpeechSeparationChallenge/`

As with the synthetic data, weights mapping speakers to microphones were generated independently from a $\mathsf{Unif}(0, 1)$ distribution, resulting in a $D \times K$ weight matrix, W. A constant "background sound level" parameter for each microphone was again added, also $\mathsf{Unif}(0, 1)$, and independent $\mathcal{N}(0, \sigma_k^2)$ noise was added to each time step at microphone k.

Results The states were inferred from the data using five models: (1) a binary-state Factorial HMM, in which the individual binary speaker sequences are modeled as independent a priori, (2) an ordinary HDP-HMM without local transitions, where the latent states are binary vectors, (3) the Sticky HDP-HMM, (4) our HDP-HMM-LT model, and (5) a model that combines the local transition property with the "Sticky" property: the Sticky HDP-HMM-LT. The same linear-Gaussian emission model was employed for all models; the only difference was the prior on the sequence of binary state vectors.

To simplify interpretation of the results, the weight matrix was fixed during inference to the true value (this makes the latent dimensions identifiable and makes comparisons between inferred and ground truth state matrices meaningful). Each model was evaluated at each Gibbs iteration using the Hamming distance between inferred and ground truth state matrices, as well as the F1 score, which is the harmonic mean between the precision (the proportion of the 1s in the inferred state matrix that were actually 1 in the ground truth) and recall (the proportion of the 1s in the ground truth state matrix that were correctly classified as 1 by the model).

The results for the five models on the synthetic and PASCAL datasets are in Figs 4.2 and 4.4, respectively. In Figs 4.3 and 4.5, the ground truth state matrix is shown against the expected state matrix for the five models, averaged over the 10 runs and all iterations 1001 through 2000. The LT and Sticky-LT models outperform the others on all measures.

The inferred decay rate λ for the HDP-HMM-LT model and its Sticky counterpart

on the synthetic and PASCAL datasets are shown in Figs. 4.2 and 4.4 as well. Both LT models settle on a non-negligible λ value for both datasets, suggesting that the local transition structure explains the data well. They also use more components than the non-LT HDP models, perhaps owing to the fact that the weaker transition prior of the non-LT models is more likely to explain nearby similar observations as a single persisting state, whereas the LT models place a higher probability on transitioning to a new state with a similar latent vector.

4.3. Synthetic Data Without Local Transitions

As a "sanity check" on the model, data was also generated directly from an ordinary HDP-HMM, with no local transition property, in order to investigate the performance of our model in a case where the data did not have the key property that its prior equipped it to discover. The results are in Fig. 4.6. During iterations when the λ parameter is large, the LT model has worse performance than the non-LT model on this data, as its bias toward local transitions is not helpful; however, the λ parameter settles near zero as sampling continues, and the model learns that a preference for local transitions does not help to explain the data. Note that when $\lambda = 0$, the HDP-HMM-LT reduces to an ordinary HDP-HMM. Unlike on the cocktail party data, the LT model does not use more states when the data does not have the LT property.

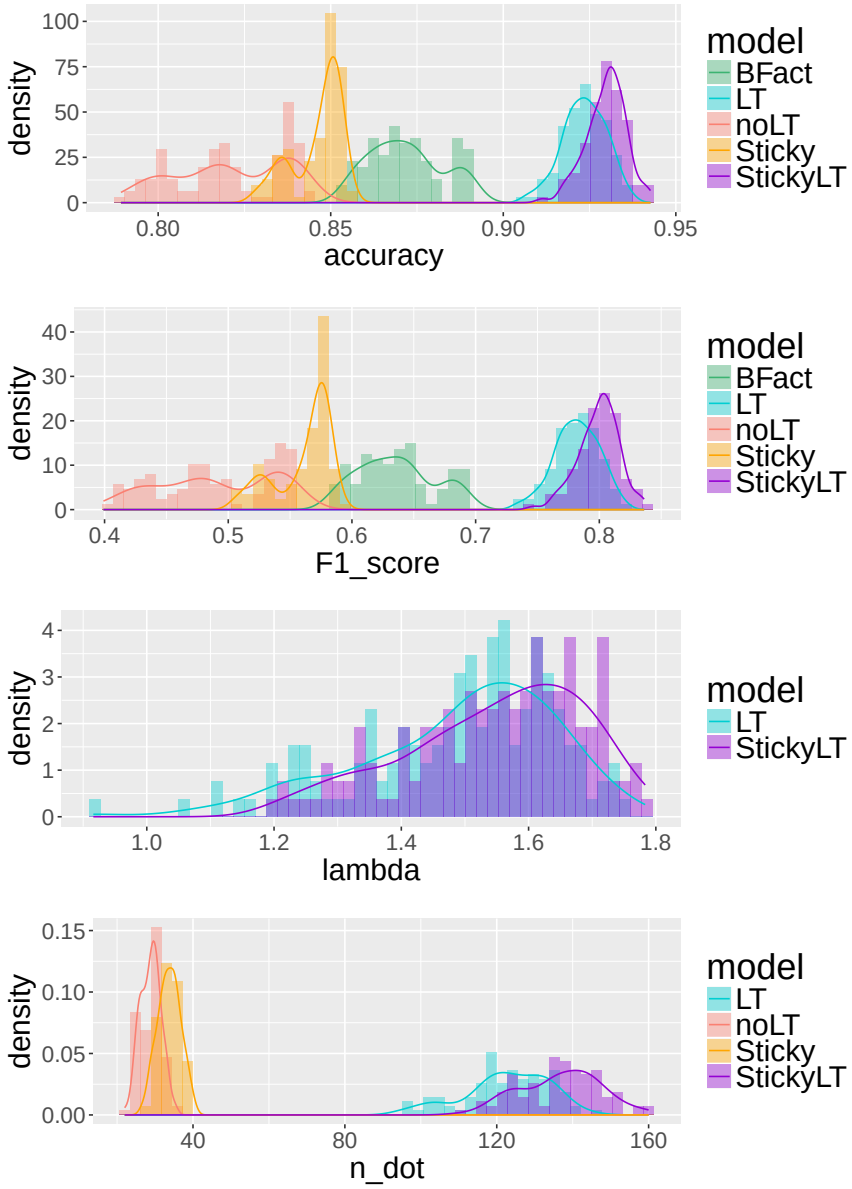


Figure 4.2: Top and next to top: Accuracy and F1 scores for the HDP-HMM-LT, standard HDP-HMM, and Binary Factorial HMM on the synthetic cocktail party data. Third from top: Learned similarity parameter, λ, for the LT model, Bottom: Number of distinct states used by HDP-HMM and HDP-HMM-LT. In all cases, iterations 1001-2000 from each of 10 Gibbs runs are shown.

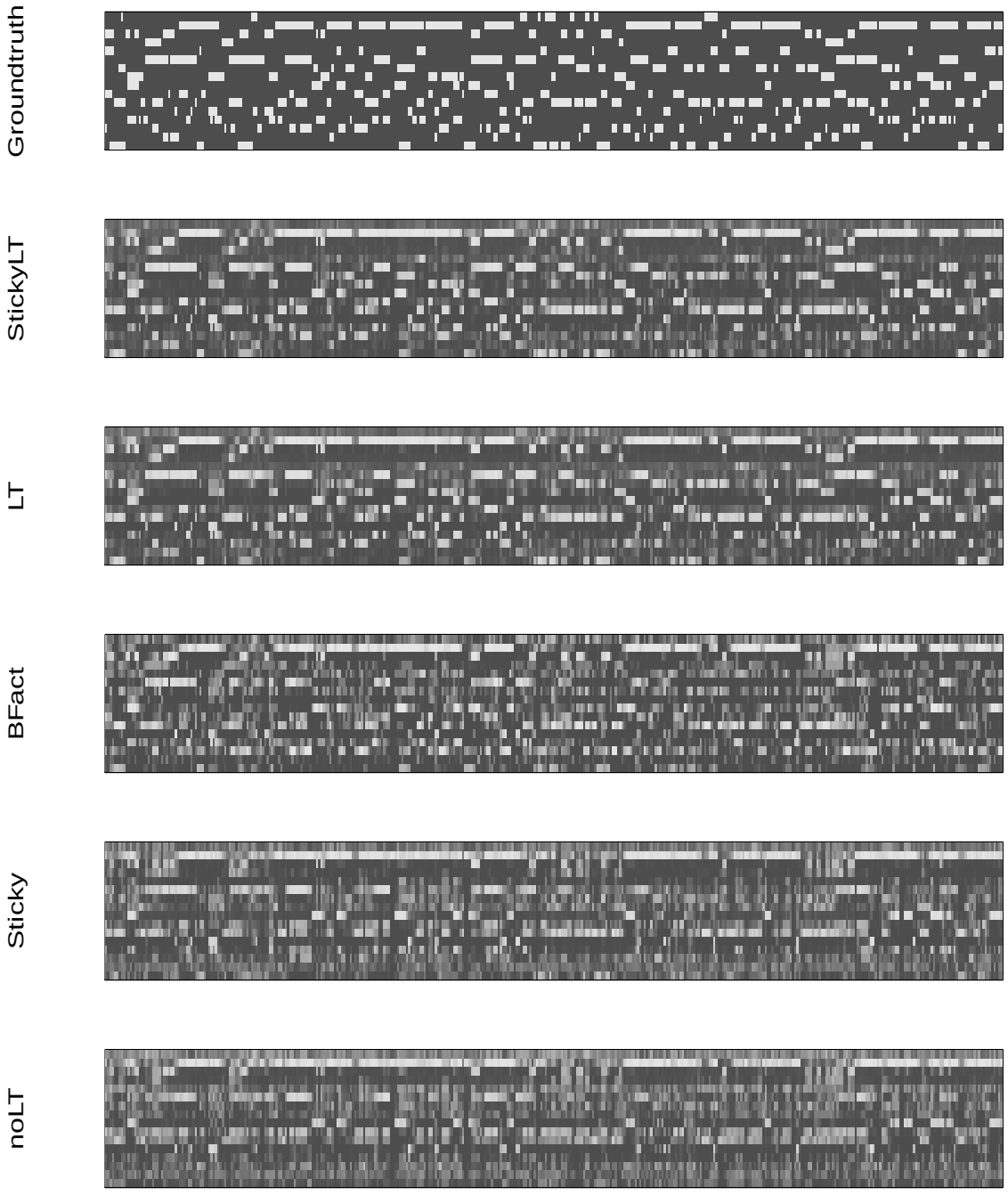


Figure 4.3: Binary speaker matrices for the synthetic cocktail data, with time on the horizontal axis and speaker on the vertical axis. White is 1, black is 0. The ground truth matrix is at the top, followed by the inferred speaker matrix for the Sticky HDP-HMM-LT, HDP-HMM-LT, binary factorial, Sticky-HDP-HMM, and "vanilla" HDP-HMM. All inferred matrices are averaged over 5 runs of 5000 Gibbs iterations each, with the first 2000 iterations discarded as burn-in.

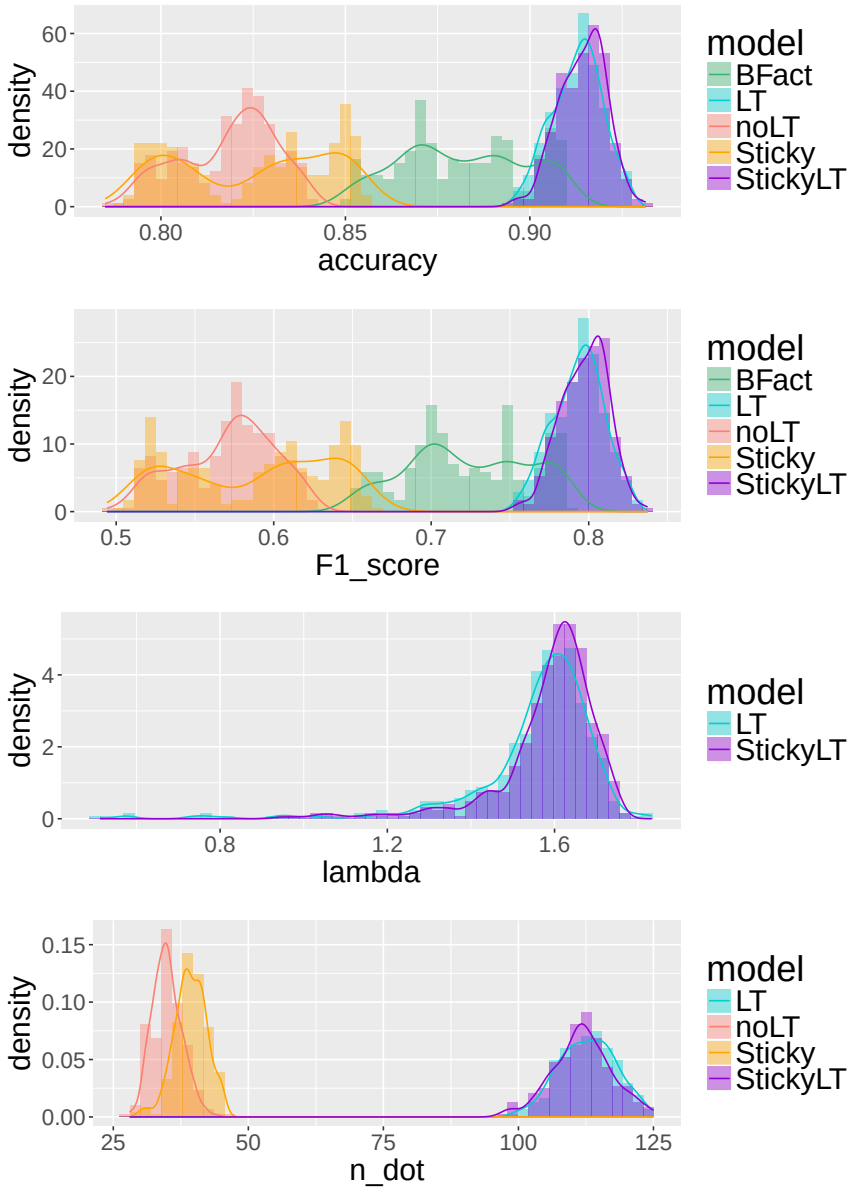


Figure 4.4: Top and next to top: Accuracy and F1 scores for the HDP-HMM-LT, standard HDP-HMM, and Binary Factorial HMM on the PASCAL cocktail party data. Third from top: Learned similarity parameter, λ, for the LT model, Bottom: Number of distinct states used by HDP-HMM and HDP-HMM-LT. In all cases, iterations 1001-2000 from each of 5 Gibbs runs are shown.

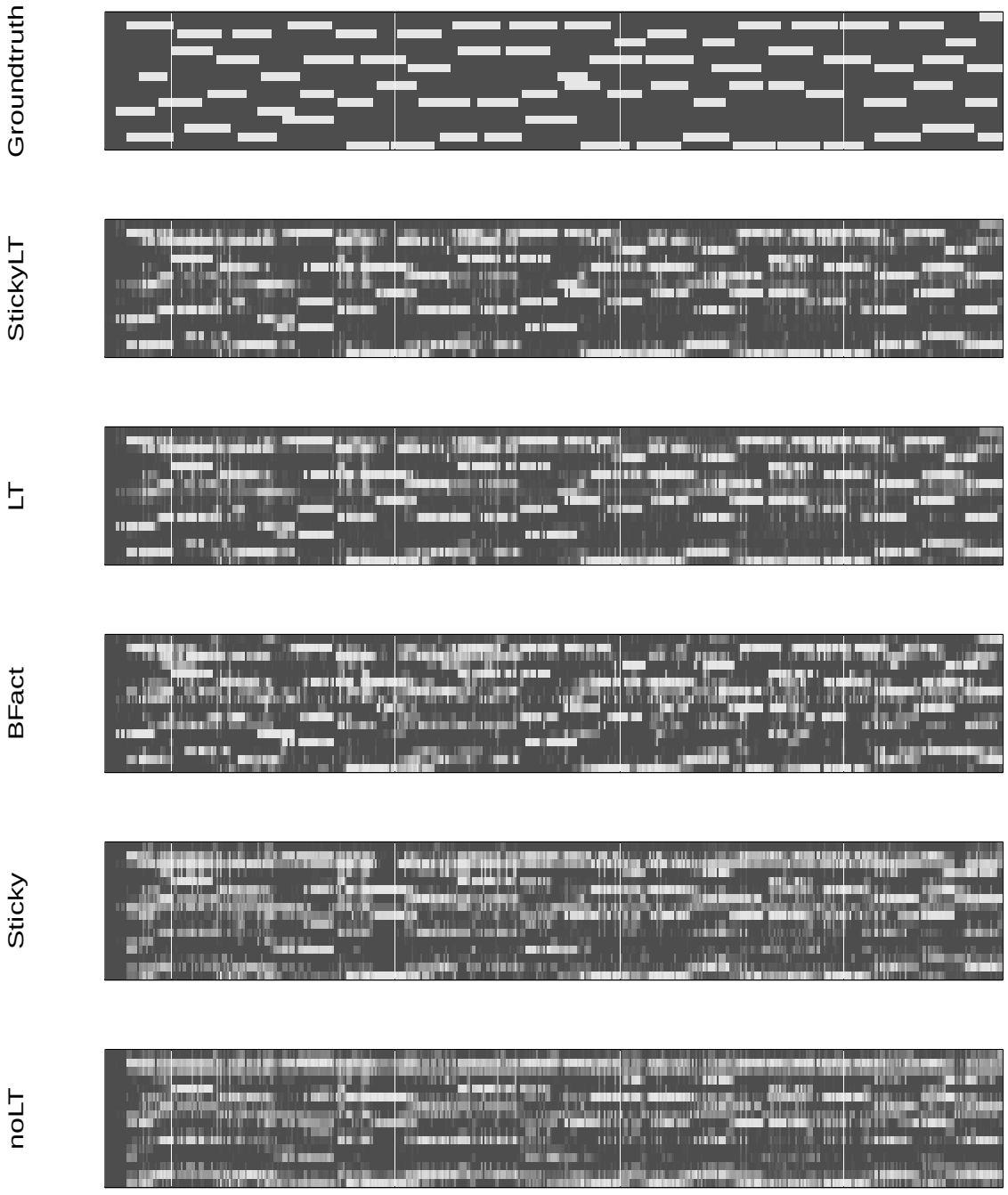


Figure 4.5: Binary speaker matrices for the PASCAL cocktail party data, with time on the horizontal axis and speaker on the vertical axis. White is 1, black is 0. The ground truth matrix is at the top, followed by the inferred speaker matrix for the Sticky HDP-HMM-LT, HDP-HMM-LT, binary factorial, Sticky-HDP-HMM, and "vanilla" HDP-HMM. All inferred matrices are averaged over 5 runs of 5000 Gibbs iterations each, with the first 2000 iterations discarded as burn-in.

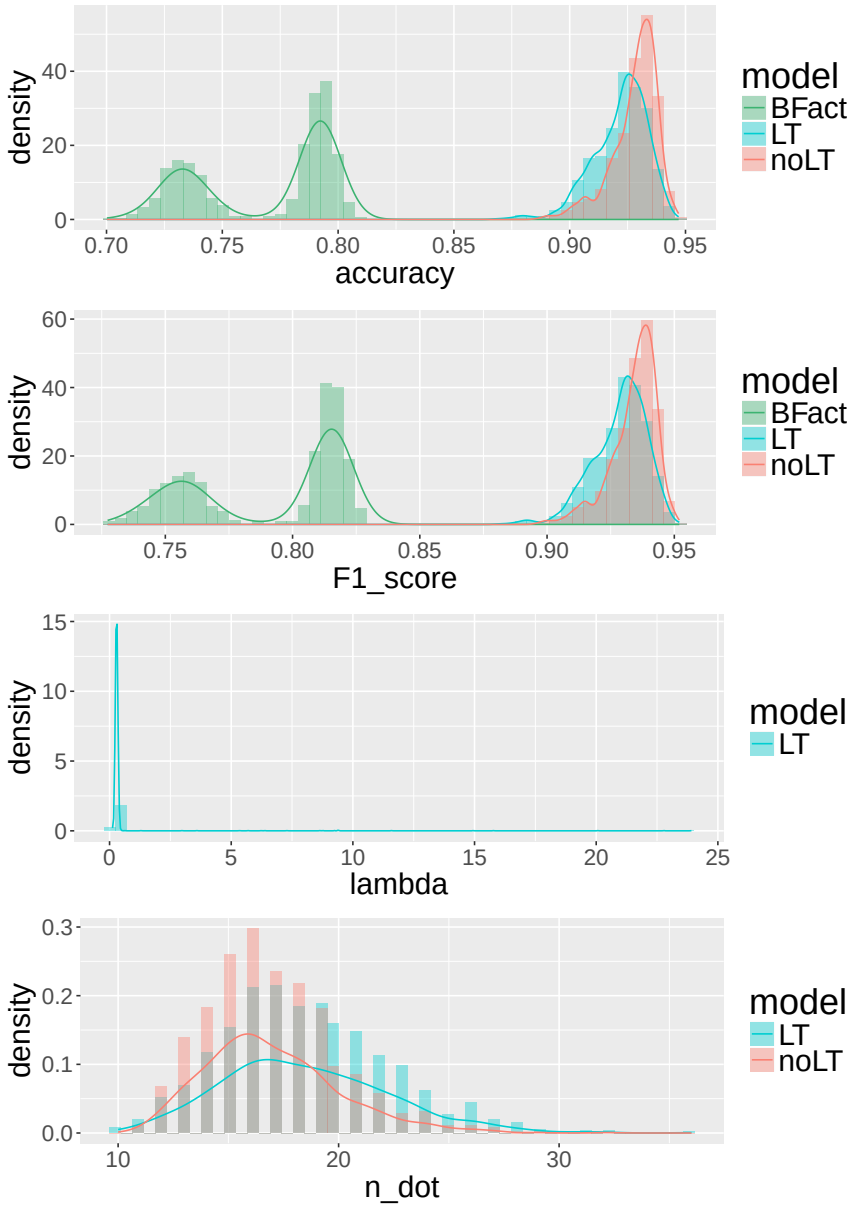


Figure 4.6: (a-b) Accuracy and F1 for the three models on data generated from an HDP-HMM without local transitions, (c) Learned similarity parameter, λ, for the LT model, (d) Number of states used by the HDP-HMM and HDP-HMM-LT. The first 1000 iterations are omitted.

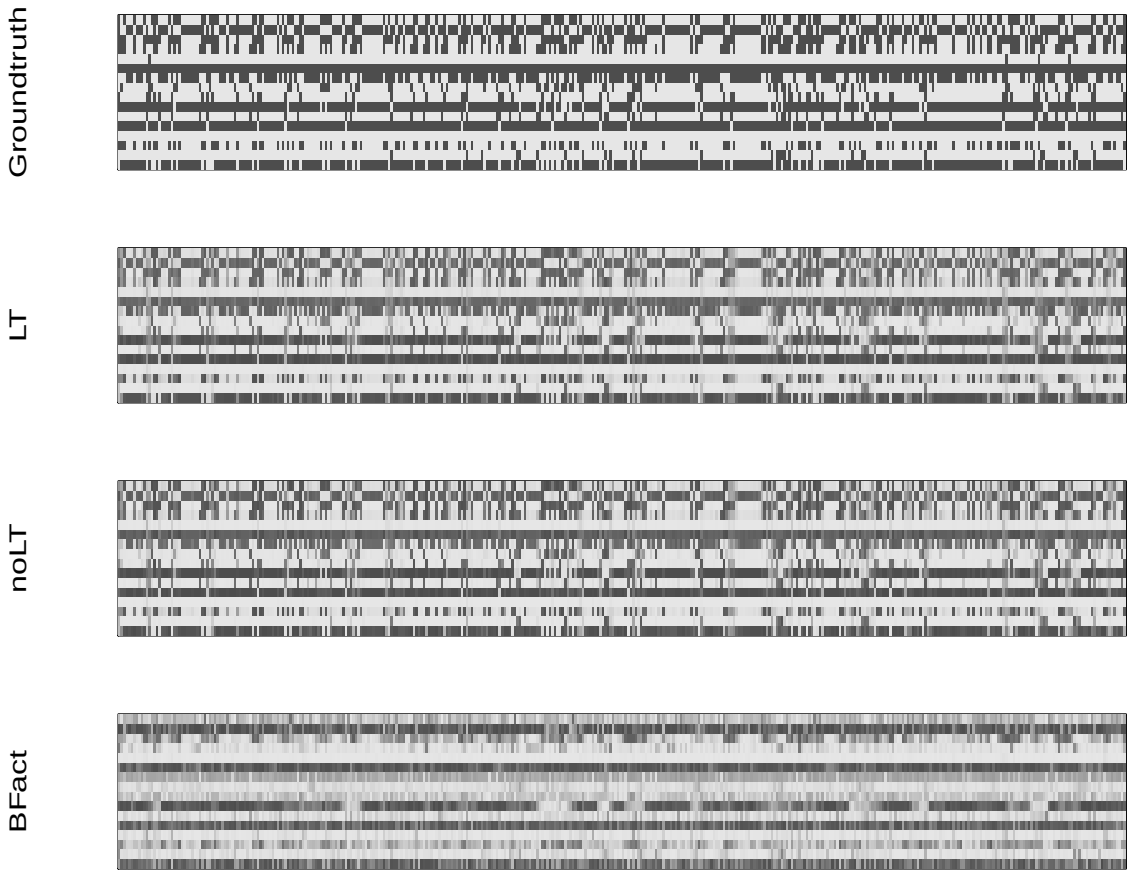


Figure 4.7: Binary state matrices for the data generated from a regular HDP-HMM, White is 1, black is 0. The ground truth matrix is at the top, followed by the inferred speaker matrix for the Sticky HDP-HMM-LT, HDP-HMM-LT, binary factorial, Sticky-HDP-HMM, and "vanilla" HDP-HMM. All inferred matrices are averaged over 5 runs of 2000 Gibbs iterations each, with the first 1000 iterations discarded as burn-in.

5. SEPARATE SIMILARITIES AND EMISSIONS: LEARNING TONAL GRAMMAR IN MUSIC

5.1. Separable Similarity and Emissions

In the HDP-HMM-LT model, we have a defined set of states with locations ℓ_j, $j = 1, \ldots, J$ and emission parameters θ_j, $j = 1, \ldots, J$. In Chapter 4, ℓ_j and θ_j were assumed to refer to a single binary state vector, and the similarity $\phi_{jj'}$ was a decreasing function of the distance between those vectors, while the emission distribution was a Gaussian centered at a linear function of θ_j.

In this chapter, I suppose instead that ℓ_j and θ_j are separate and a priori independent, where the θ_j govern the emission distributions, and the similarities, $\phi_{jj'}$ depend on the ℓ_j

Define

$$\phi_{jj'}(\ell_j, \ell_{j'}) = \exp\left(-\frac{\lambda}{2}\Delta^2_{jj'}\right)$$

where $\Delta_{jj'}$ is the Euclidean distance between ℓ_j and $\ell_{j'}$; that is,

$$\Delta^2_{jj'} = \sum_d (\ell_{jd} - \ell_{j'd})^2$$

5.2. A Hamlitonian Monte Carlo Step to Sample ℓ

Since the ℓ_j are real-valued vectors, we can sample them jointly using Hamiltonian Monte Carlo (HMC, also known as "Hybrid Monte Carlo"; Duane et al. (1987), see also Neal et al. (2011)). HMC is a variation on the Metropolis-Hastings MCMC algorithm which is designed to more efficiently explore a high-dimensional continuous distribution by adopting a proposal distribution which is based on the evolution of Hamiltonian dynamics in a physical system. The position of the particle in the system represents the current state of the Markov chain, the potential energy of the particle

is the negative log of the target distribution, and an auxiliary "momentum" variable is introduced, representing the kinetic energy of the system. The Markov chain evolves by computing a discrete approximation of an update to the position and momentum variables, and then computing the standard Metropolis-Hastings acceptance probability.

In order to carry out HMC in the context of the HaMMLeT model with latent continuous state variables given by ℓ_j, $j = 1, \ldots, J$, we need the log likelihood and log prior for the ℓ vector. Assume independent and isotropic Gaussian priors on each ℓ_j, so we have

$$p(\ell_j) \propto \exp\left(-\frac{h_\ell}{2}\sum_d \ell_{jd}^2\right),$$

where h_ℓ is the prior precision which does not depend on d.

Then the log prior density, up to a constant c, is

$$\log p(\ell_j) = c - \frac{h_\ell}{2}\sum_d \ell_{jd}^2$$

The relevant log likelihood, as shown in Chapter 3 is the probability of the z and Q variables given the $\phi_{jj'}$. In particular, we have

$$L := p(z, Q \mid \phi) \propto \prod_j \prod_{j'} \phi_{jj'}^{n_{jj'}} (1 - \phi_{jj'})^{q_{jj'}}$$

where factors not containing ϕ are absorbed into the proportionality constant, and

$$\log L = c_2 + \sum_j \sum_{j'} \left(n_{jj'} \log(\phi_{jj'}) + q_{jj'} \log(1 - \phi_{jj'})\right)$$

To do Hamiltonian Monte Carlo to sample from the conditional posterior of ℓ given $\mathbf{z}$ and $\mathbf{Q}$, we need to compute the gradient of the log posterior, which is just the sum of the gradient of the log prior and the gradient of the log likelihood.

The j, d coordinate of the gradient of the log prior is simply $-h_\ell \ell_{jd}$.

To get the j, d coordinate of the gradient of the log likelihood, we can apply the chain rule to terms as is convenient. In particular,

$$\frac{\partial L}{\partial \ell_{jd}} = \sum_j \sum_{j'} n_{jj'} \frac{\partial \log(\phi_{jj'})}{\partial \Delta^2_{jj'}} \frac{\partial \Delta^2_{jj'}}{\partial \ell_{jd}} + \sum_j \sum_{j'} q_{jj'} \frac{\partial \log(1-\phi_{jj'})}{\partial (1-\phi_{jj'})} \frac{\partial (1-\phi_{jj'})}{\partial \Delta^2_{jj'}} \frac{\partial \Delta^2_{jj'}}{\partial \ell_{jd}}$$

We have the following components:

$$\begin{aligned}
\frac{\partial \log(\phi_{jj'})}{\partial \Delta^2_{jj'}} &= -\frac{\lambda}{2} \\
\frac{\partial \Delta^2_{jj'}}{\partial \ell_{jd}} &= 2\Delta_{jj'd} I(j \neq j') \\
\frac{\partial \log(1-\phi_{jj'})}{\partial (1-\phi_{jj'})} &= \frac{1}{1-\phi_{jj'}} \\
\frac{\partial (1-\phi_{jj'})}{\partial \Delta^2_{jj'}} &= \frac{\lambda}{2}\phi_{jj'}
\end{aligned}$$

which yields

$$\begin{aligned}
\frac{\partial L}{\partial \ell_{jd}} &= -\lambda \sum_j \sum_{j'} n_{jj'} \Delta_{jj'd} \mathbb{I}(j \neq j') + \lambda \sum_j \sum_{j'} q_{jj'} \Delta_{jj'd} \frac{\phi_{jj'}}{1-\phi_{jj'}} \mathbb{I}(j \neq j) \\
&= -\lambda \sum_{(j,j'):j \neq j'} \Delta_{jj'd} \left(n_{jj'} - q_{jj'} \frac{\phi_{jj'}}{1-\phi_{jj'}} \right)
\end{aligned}$$

5.3. Discovering Chord Equivalence Classes in Tonal Music

A real-world test of the separable-similarity form of HaMMLeT comes from the music domain. Two datasets were used, one generated from a an artificial musical grammar, the other based on Bach chorales. In both cases, each observation consisted of four musical tones played concurrently, constituting a chord, and the sequences between chords followed conventions for Western tonal music.

5.3.1. Emission model

The emission model used for the music data is Dirichlet-Categorical, with each state modeled using a categorical distribution over the 120 integer chord symbols, with a symmetric Dirichlet prior on the emission distributions. In the experiments reported here the concentration parameter was 1.0.

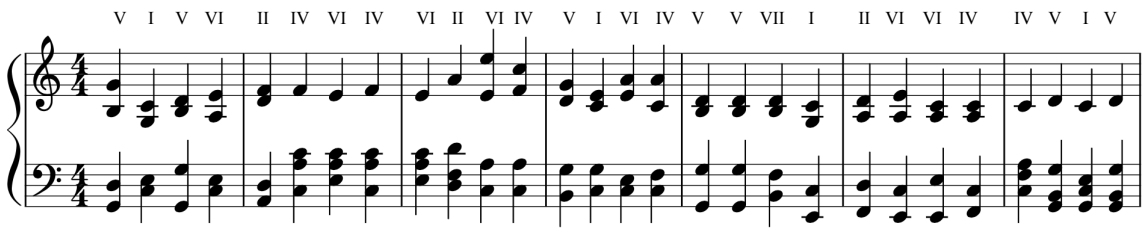


Figure 5.1: Transcription of a segment of a four-voice chorale produced by Kulitta, annotated with chord equivalence classes. The full data sequence was 500 chords long.

5.3.2. Synthetic data

Data-Generation To create the first dataset, two 500-chord sequences were generated by the Kulitta generative grammar (Quick, 2014), one used as a training set and the other used as a test set. A score from an excerpt of the training sequence is in Fig. 5.1. The chord sequences were preprocessed for the model by translating each chord to a single integer value, with each distinct set of four tones receiving a distinct label. In total, throughout the two 500-chord sequences, 120 distinct chords appeared, 13 of which were unique to the test set. The sparsity of the observed data suggests that a more compact latent state representation might be useful. Of interest is whether the models can discover such a representation, grouping the observed chords into classes with a low degree of overlap and a high degree of functional interchangeability.

Results In this data there is no ground truth latent state sequence to compare the results to, so instead of accuracy measures, marginal likelihood on training and test sets was used as a performance metric, computed at each iteration using forward message-passing. Results are in Fig. 5.2. Unlike on the cocktail party data in Chapter 4, here the LT model does not show a likelihood advantage on the training data (in fact, its marginal likelihood is lower); however, the LT model performs consistently better at generalizing to the test set. This suggests that the more informative prior on the transition matrix is helping to extract stable structure, and avoid overfitting

the statistics of the training set.

It is also of interest to examine the set of latent states discovered by the two models, and in particular the degree to which the stochastic map between latent states and produced chords approximates a one-to-many structure. The emission distributions for the last iteration of the first run of each model are shown in Fig. 5.3. Notably, the LT model does a better job of partitioning chord symbols, with fewer instances of a single chord being generated with relatively high probability in more than one state, as can be seen by examining how often a column (representing a chord) appears with high probability (bright squares) in more than one latent state (rows).

5.3.3. Bach chorales

A second music experiment used the same model but evaluated it using a real, and much larger dataset, namely a corpus of 217 four-voice major key chorales by J.S. Bach from music21[1].

Data Two-hundred chorales from the corpus were randomly selected as a training set, with the other 17 used as a test set to evaluate surprisal (marginal log likelihood per observation) by the trained models. All chorales were transposed to C-major, and each distinct four-voice chord (with voices ordered) was encoded as a single integer. In total there were 3307 distinct chord types and 20401 chord tokens in the 217 chorales, with 3165 types and 18818 tokens in the 200 training chorales, and 143 chord types that were unique to the test set.

Since the chords were encoded as integers, the emission distribution for each state is $\mathsf{Categorical}(\theta_j)$. We use a 3307-dimensional symmetric Dirichlet prior for each θ_j, resulting in conjugate updates to θ conditioned on the latent state sequence, z.

[1] `http://web.mit.edu/music21`

The prior on each ℓ was bivariate $\mathcal{N}(0, \mathbf{I}_2)$, and the similarity function was squared exponential based on Euclidean distance: $\phi_{jj'} = \exp\{-\lambda d_2(\ell_j, \ell_{j'})^2\}$ where d_2 is Euclidean distance.

Results Four models: HDP-HMM-LT, Sticky-HDP-HMM-LT, HDP-HMM and Sticky-HDP-HMM, were evaluated on the Bach data using 5 Gibbs chains running for 10,000 iterations each on the 200 training chorales, which were modeled as conditionally independent of one another. Every 50th iteration, the marginal log likelihood was calculated on the 17 test chorales (integrating out z). The distributions of the training and test log likelihoods are in Fig. 5.4. Although the LT model does not achieve as close a fit to the training data, its generalization performance is better, suggesting that the HDP-HMM without the local transition property is overfitting. In fact, the vanilla HDP-HMM and Sticky HDP-HMM are maxing out their state allowances at 200, whereas the LT models average closer to 150 states. It is perhaps counterintuitive that the LT property would protect against overfitting, as it has more free parameters. However, the similarity bias induces greater sharing of information across parameters, as in a hierarchical model: instead of each entry $\pi_{jj'}$ being informed only by transitions from j to j', it is informed to some extent by *all* transitions, as all transitions inform the similarity structure.

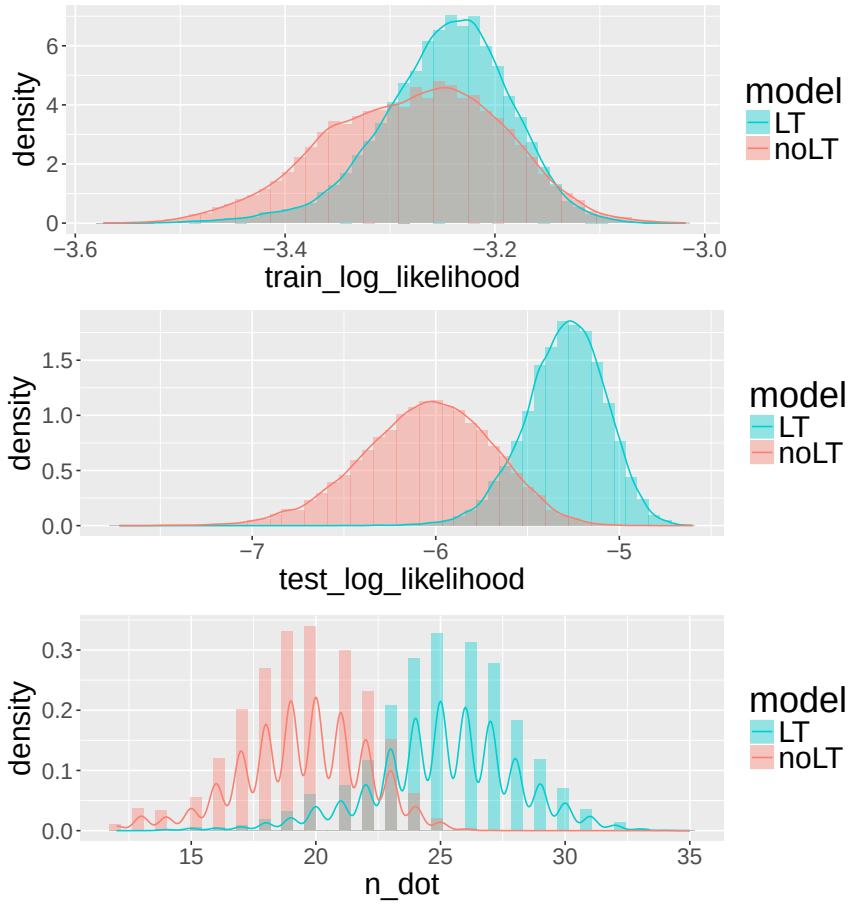


Figure 5.2: Top: Log likelihood on the Kulitta training set, aggregating over 10 Gibbs runs of 5000 iterations each, and omitting the first 2500 iterations as burn-in. For all log likelihood measures, the state sequence, z, has been marginalized out using forward message-passing. Middle: Log likelihood on a held out test set generated by the same Kulitta grammar. Bottom: the number of occupied states in the training data.

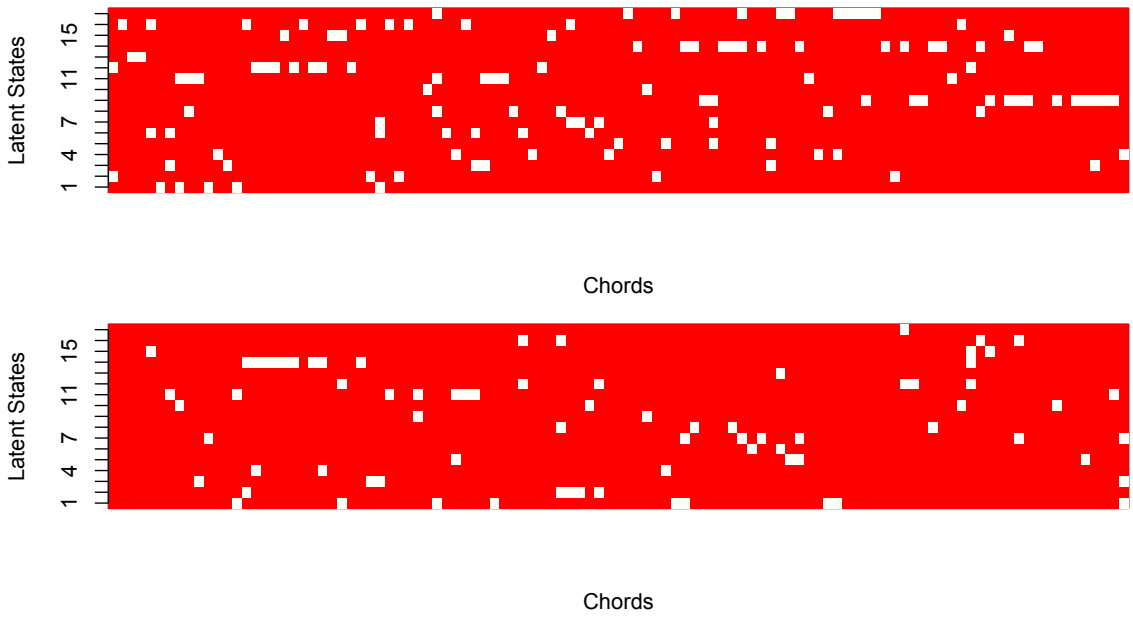


Figure 5.3: Emission distributions of the states discovered by the HDP-HMM (top) and the HDP-HMM-LT (bottom). Rows correspond to latent states actually visited during either the training or test sequence; columns represent the distinct chord symbols emitted. Emission probabilities are thresholded at 3 times the relative frequency of a chord in the corpus, so that bright spots correspond to a higher-than-average probability of emitting a particular chord, irrespective of how common or rare the chord is.

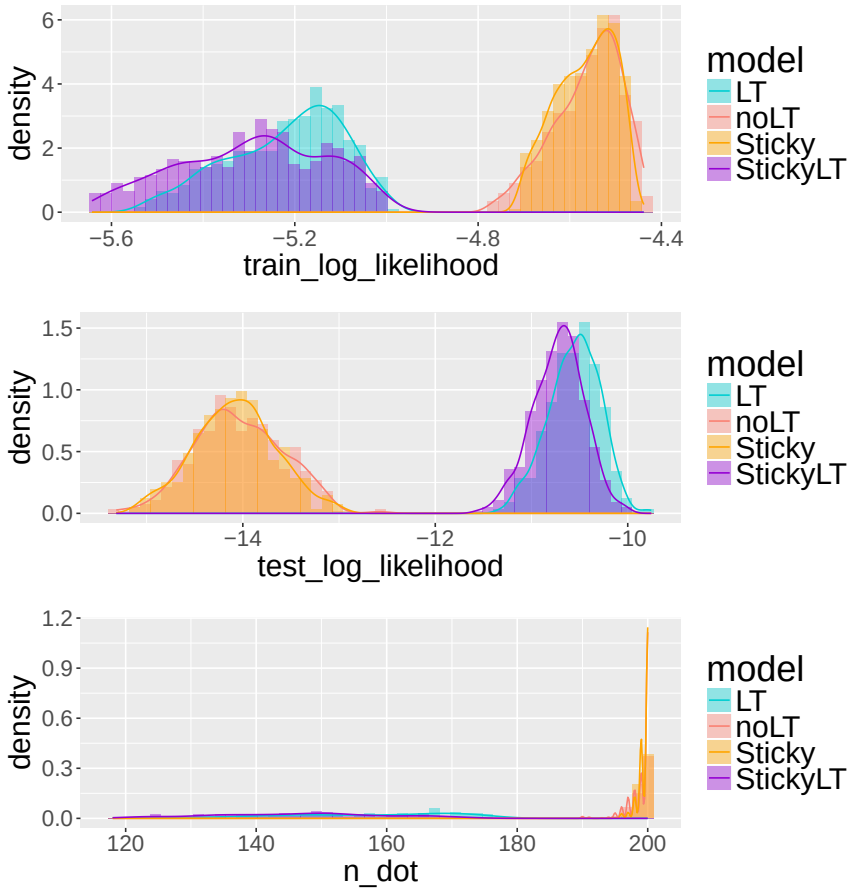


Figure 5.4: Top and Middle: Training set and test set log marginal likelihoods for Bach chorale data on the four HDP-based models: HDP-HMM-LT, HDP-HMM, Sticky HMM, and Sticky HDP-HMM-LT. Bottom: Number of latent states occupied in the training set by each model.

6. CONCLUSIONS AND FUTURE WORK

6.1. Summary

In this dissertation I have presented a new probability model for sequence data: the Hierarchical Dirichlet Process Hidden Markov Model with Local Transitions (HDP-HMM-LT, or "HaMMLeT"), which provides a way to infer a latent state sequence underlying time series data while incorporating prior information that the latent state space is structured in such a way that transitions are more likely between states that are "nearby" or similar in some sense.

Two broad versions of the model are presented, which represent similarity fundamentally differently. The first version of the model, illustrated in Chapter 4 conceives of similarity as a function of the emission distributions of the respective states; or at least, of a function of the same state information that influences the emission distributions. This form of the model is well suited to scenarios in which the latent state is a composite of several component sub-states, not all of which change at the same time; that is, the kinds of settings where factorial models are typically used. The difference between the HaMMLeT model and a factorial model is that the sub-states are not modeled as independent sequences in HaMMLeT, and so it is possible to discover correlations among them, and in particular to discover a relatively small set of global states that represent high probability joint configurations of the component chains. The performance of this model is illustrated in a speaker diarization experiment (the "cocktail party" problem), with favorable comparison to both the "vanilla" HDP-HMM and the factorial HMM, whose features it combines, as well as to the Sticky HDP-HMM, which privileges self-transitions but possesses no notion of proximity between non-identical states.

The second version of the model, illustrated in Chapter 5, represents similarity

between states as unrelated to emissions themselves, which is suited to applications where there is a notion of "functional distance" between states which is separate from the surface distance between observations at those states, as is arguably the case in Western music, where the "circle of fifths" structures harmonic classes in a manner such that consecutive chords tend to be nearby in the circle, even if their pitches are far apart. This model is illustrated using two harmonic parsing experiments, one using synthetically generated data from the Kulitta grammar Quick (2014), and the other using Bach chorales. Comparisons are made to the vanilla HDP-HMM and the Sticky HDP-HMM. Here, the key finding is that although the two "non-LT" models fit the training data well (in fact better than the LT model), they appear to overfit, yielding worse predictive likelihood on a held out test set than the LT model, which evidently recovers a more parsimonious description of the harmonic structure.

I will conclude the dissertation by describing some directions for future work extending and further exploring the properties of the HaMMLeT model. First I will describe a potential application related to power disaggregration, which has been a use case for existing nonparametric Bayesian HMMs (e.g., Johnson and Willsky (2013)). Then I describe a possible marriage between the local transition property of HaMMLeT with the nonparametric infinite factorial HMM Gael et al. (2009). Finally, I discuss computational challenges for inference with the HaMMLeT model, and potential directions for improving scalability.

6.2. Future Application: Power Disaggregation

A potential application of the categorical-vector-state model described in Chapter 4 is to disaggregating energy signals. In the future, I plan to test the model using a subset of the Reference Energy Disaggregation Data set (REDD; Kolter and Johnson (2011)). The data consists of power consumption measurements from several time intervals from several electrical channels in several different homes, as well as an

aggregated power consumption time series for each interval per home. An example interval is shown in Fig. 6.1.

Two key properties of this data are apparent in the figure. First, up to some small-scale variability, each channel visits a relatively small number of amplitudes, with some (such as the oven) alternating between an "on" state and an "off" state, and others (such as the outlets and lighting channels) exhibiting more complex dynamics, presumably corresponding to various appliances or light fixtures turning on and off in different combinations. This property motivates the choice of a categorical state model. Second, there are clear correlations among the various channels: in this example, the two oven channels always go on and off together, albeit exhibiting different amplitudes when on, while two of the three lighting channels are highly but not perfectly correlated, while the third exhibits opposing behavior to the other two (perhaps arising from occupants moving between areas of the house and turning off lights as they leave a room).

Collectively, the dynamics exhibited here are similar to the cocktail party data described in Chapter 4. For one, since transitions between combinatorial states tend to be "local", in that it is rare for more than one channel to change state at one time, we would expect the HaMMLeT model's bias toward local state transitions to be beneficial, compared to the "vanilla" HDP-HMM which has no such locality bias. On the other hand, not all combinations of states occur in the data, and morever, some state combinations are more or likely than the product of the individual channel probabilities would suggest, suggesting that the flexibility of a model such as HaMMLeT, in which there is a single latent state space over vector-valued states, might be better able to capture the correct transition probaiblities than a model such as the Factorial HMM, in which the component chains evolve *a priori* independently.

6.2.1. Generalizing to categorical-valued θ

It is not difficult to relax the assumption made in Chapter 4 that the θ_j are binary vectors and instead allow each θ_{jd} to take on one in an arbitrary set of discrete values — i.e., to be categorically rather than Bernoulli distributed. The similarity between two states could again be defined in terms of distance between θ_j vectors that also inform the emission distribution via an additive model, perhaps also linear-Gaussian. As a result, many of the additional inference steps described for the cocktail party experiment would carry over to this instantiation of the model. The exception of course is inferences about θ.

In this section, I define representations and priors for θ and the weight matrix, W for the general case of categorical-vector-valued θ. Then I derive the necessary Gibbs steps.

A potential future application of this version of the model to power disaggregation is described above, where the observations consist of an aggregated energy-use time series from several houses, the latent state vector describes the energy being used by each of several channels (e.g., lighting, refrigerator, washer/dryer), such that individual channels are in one of several discrete states at a given time. For example, the refrigerator channel might cycle between an "off" state, a low-power standby state, and a high-power "active cooling" state. Inference consists of discovering the set of discrete states for each channel, attributing a typical level of energy consumption for each state, and inferring what state each channel is in during each discretized time window in the data.

6.2.2. Priors and representations in the categorical state variant

As in the binary vector model described in Chapter 4, we will assume that both ℓ_j and θ_j refer to a single vector state description, but instead of binary elements, the elements are arbitrary integer-valued. In place of Beta priors on each ℓ_{jd} (aka θ_{jd}),

we can use Chinese Restaurant Process priors, with concentration parameter $\alpha_d^{(\ell)}$ for channel d. In the case that the number of categorical values is known in advance, this can be replaced by a Dirichlet prior, but I present the more flexible case here.

Similarity model I will continue to assume that the ϕ function decays exponentially as a function of the number of component-wise differences (that is, Hamming distance) between a pair of states. Specifically, $\Delta_{jj'd}$ is zero if and only if $\ell_{jd} = \ell_{j'd}$, and is 1 otherwise, and we define $\Delta_{jj'} = \sum_d \Delta_{jj'd}$, and $\phi(\ell_j, \ell_{j'}) = e^{-\lambda \Delta_{jj'}}$.

Emission model Turning to the emission model, I will use the θ_j notation in place of the ℓ_j notation.

We can use a "dummy variable" representation of θ_j to facilitate the definition of a linear model. Define S_d to be the number of realized states for dimension d and a 1 in position $\sum_{d'<d} S_{d'} + s$ indicates that $\theta_{jd} = s$. There will thus be D entries equal to 1, with the remaining entries equal to zero. We can then represent the weight matrix W as stacked block matrix, where each block is $S_d \times K$, and there is one block for each d. In practice we only need to instantiate a new block when some θ_{jd} is assigned to a "new table" in the CRP metaphor, so that the dimension of $\mathbf{W}$ is $\sum_d S_d \times K$. Then we have

$$y_t \sim \mathcal{N}(w^{(b)} + W^{\mathsf{T}} \theta_{z_t}, \Sigma) \tag{6.1}$$

where $w^{(b)}$ is a K-dimensional bias vector with a separate Normal prior, W is the weight matrix as defined just above, z_t is the state indicator for time t, and Σ is a $K \times K$ noise covariance matrix.

6.2.3. Adapting posterior inference for categorical state vectors

Sampling θ (aka ℓ) As before, the conditional posterior for θ_{jd} is proportional to the product of three terms: the prior (now a CRP), the likelihood of all successful and failed transitions to and from state j, and the likelihood of the observation sequence.

Under the CRP, the probability that $\theta_{jd} = s$ conditioned on the rest of θ (but not the data) is proportional to the number of other $j' \neq j$ such that $\theta_{j'd} = s$ where this count is positive; and proportional to $\alpha_j^{(\theta)}$ otherwise. Let

$$\tilde{n}_{ds}^{-j} = \sum_{j' \neq j} I(\theta_{jd} = s), \quad s = 1, \ldots, S_d \tag{6.2}$$

be these counts, where we assume that there are S_d distinct values taken by the θ_{jd} for a particular d. Then

$$p(\theta_{jd} = s \,|\, \theta_d^{-j}) \propto \begin{cases} \tilde{n}_{ds}^{-j} & s = 1, \ldots, S_d \\ \alpha_d^{(\theta)} & s = S_d + 1 \end{cases} \tag{6.3}$$

The transition component of the likelihood is as in the binary case:

$$p(z, Q \,|\, \theta_{jd}, \theta_d^{-j}) \propto e^{-\lambda \sum_{j'} \Delta_{jj'}(n_{jj'} + n_{j'j})} \prod_{j' \neq j} (1 - e^{-\lambda \Delta_{jj'}(q_{jj'} + q_{j'j})}) \tag{6.4}$$

$$\propto e^{-\lambda \sum_{j'} I(\theta_{jd} \neq \theta_{j'd})(n_{jj'} + n_{j'j})} \prod_{j' \neq j} (1 - a \cdot e^{-\lambda I(\theta_{jd} \neq \theta_{j'd})(q_{jj'} + q_{j'j})}) \tag{6.5}$$

where a is a constant in θ_{jd}, defined as $e^{-\lambda \Delta_{jj'-d}(q_{jj'd} + q_{j'jd})}$. Taking a log yields

$$\log p(z, Q \,|\, \theta_{jd}, \theta_d^{-j}) = -\lambda \sum_{\{j' :\ \theta_{jd} \neq \theta_{j'd}\}} (n_{jj'} + n_{j'j}) + \sum_{j' \neq j} \log(1 - a \cdot e^{-\lambda I(\theta_{jd} \neq \theta_{j'd})(q_{jj'} + q_{j'j})}) \tag{6.6}$$

The emission component of the likelihood is given for each t by

$$p(y_t \,|\, \theta_{z_t}, W, \Sigma) \propto |\Sigma|^{-1/2} \exp(-\frac{1}{2}(y_t - w^{(b)} - W^{\mathsf{T}} \theta_{z_t})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (y_t - w^{(b)} - W^{\mathsf{T}} \theta_{z_t})) \tag{6.7}$$

Assuming a diagonal covariance matrix, isolating $\theta_{j,d}$, and taking a log, this becomes, for each k and t,

$$\log p(y_{tk} \,|\, \theta_{z_t,d}, \theta_{z_t}^{-d}, w_k^{(b)}, \sigma_k^2) = -\frac{1}{2\sigma_k^2} \left(y_{tk} - w_k^{(b)} - \theta_{z_t} \mathbf{w}_k \right)^2 \tag{6.8}$$

$$\propto -\frac{1}{2\sigma_k^2} (\sum_{d'} w_{\theta_{z_t d'}, d', k} - (y_{tk} - w_k^{(b)}))^2 \tag{6.9}$$

$$\propto -\frac{1}{2\sigma_k^2} (w_{\theta_{z_t d}, d, k} - (y_{tk} - w_k^{(b)} - \sum_{d' \neq d} w_{\theta_{z_t, d'}, d', k}))^2 \tag{6.10}$$

For a particular j and d, the full emission log likelihood is a sum of terms like the above, over all t such that $z_t = j$.

Sampling W Having expanded θ to a dummy variable representation, and having constructed a stacked block form of W, we can sample each column of W from a conditional posterior multivariate Normal just as in the binary case, with new columns added during inference being sampled from the prior.

Sampling λ Since λ controls only the relationship between the distance matrix, Δ, and the similarity matrix, ϕ, upon conditioning on Δ and the jump attempts, λ is not sensitive to the way that Δ was computed; hence it can be sampled using Adaptive Rejection Sampling from exactly the same conditional distribution as before.

6.3. Scaling Inference Using Beam Sampling

A limitation of the weak limit approximation defined in this dissertation and used to carry out Gibbs sampling in all experiments discussed is that requires a maximum number of states (referred to as J in the text) to be specified in advance. Although J can be set as large as desired to approximate a truly infinite state model, and the hierarchical structure of the HDP prior ensures that a sparse subset of available states will be used, the inference algorithm scales quadratically in J (specifically, on the order of $\mathcal{O}(TJ^2)$, where T is the number of observations), due to the multiplication by the transition matrix that must occur during the forward message passing step. Hence it may be quite costly to perform inference if a very large J is needed. Because of this, inference algorithms that scale better and, ideally, eliminate entirely the need for the upper bound on the size of the state space.

A promising possibility comes from **beam sampling** (Van Gael et al., 2008), which in the ordinary HDP-HMM allows the z_t to be sampled jointly from the full infinite state HDP model without evaluating all state combinations at every t. This is achieved through **slice sampling** (Neal, 2003). Slice sampling is a Gibbs sampling method in which the goal of sampling values x from a target density, $f(x)$ is achieved

by the introduction of an auxiliary "slice" variable, s that depends on x according to the density $f(s \mid x)$, chosen so that $f(x \mid s)$ can be sampled from easily. By the usual logic of a Gibbs sampler, in the limit we obtain samples from the joint distribution $f(x, s)$, the x-marginal of which is $f(x)$.

6.3.1. Beam sampling in the original HDP-HMM

Beam sampling (Van Gael et al., 2008) combines slice sampling with the forward-backward algorithm as follows. The original conditional distribution of each z_t is given by the transition matrix:

$$p(z_t = j' \mid z_{t-1} = j, \pi) = \pi_{jj'} \tag{6.11}$$

For each t we introduce a slice variable, s_t, which is uniformly distributed on the interval $(0, \pi_{z_{t-1}z_t})$, so that we have the conditional density

$$p(s_t \mid z, \pi) = \pi_{z_{t-1}z_t}^{-1} \mathbb{I}(0 < s_t < \pi_{z_{t-1}z_t}) \tag{6.12}$$

This yields the joint density

$$p(z, s \mid \pi) = \prod_{t=1}^{T} \pi_{z_{t-1}z_t} \pi_{z_{t-1}z_t}^{-1} \mathbb{I}(0 < s_t < \pi_{z_{t-1}z_t}) \tag{6.13}$$

$$= \prod_{t=1}^{T} \mathbb{I}(0 < s_t < \pi_{z_{t-1}z_t}), \tag{6.14}$$

that is, conditioned on the collection of slice variables (but not yet conditioned on the data), the distribution of state sequences is uniform over all sequences that meet the condition that $\pi_{z_{t-1}z_t} > s_t$ for all t.

Since for each j, the transition probabilities, $\pi_{jj'}$ must sum to 1 over all j', there can be only a finite number of choices for j' at each t that satisfy $\pi_{jj'} > s_t$; hence, conditional on s, we need only consider finitely many states during the forward and backward steps.

Now, conditioning on the data y as well, and defining $b^*_{t+1}(j') := p(z_{t+1} = j' \mid s, y_1, \ldots, y_{t+1})$, the forward message passing recurrence relation becomes

$$b^*_{t+1}(j') = p(z_{t+1} = j' \mid s, y_1, \ldots, y_{t+1}) \tag{6.15}$$

$$\propto p(z_{t+1} = j', s, y_1, \ldots, y_t) p(y_{t+1} \mid z_{t+1} = j') \tag{6.16}$$

$$= p(y_{t+1} \mid z_{t+1} = j') \sum_j p(z_t = j, z_{t+1} = j', s, y_1, \ldots, y_t) \tag{6.17}$$

$$= p(y_{t+1} \mid z_{t+1} = j') \sum_j p(z_t = j, s, y_1, \ldots, y_t) p(z_{t+1} = j' \mid z_t = j, s) \tag{6.18}$$

$$\propto f(y_{t+1} \mid \theta_{j'}) \sum_j b^*_t(j) I(\pi_{jj'} > s_{t+1}) \tag{6.19}$$

Although the sum over j is still technically over an infinite number of terms, we can show that only finitely many j will have $b^*_t(j) > 0$.

First, consider b^*_1. We have

$$b^*_1(j') \propto f(y_1 \mid \theta_{j'}) I(\pi_{0j'} > s_t). \tag{6.20}$$

Since only finitely many entries in the initial distribution π_0 can exceed s_t, $b^*_1(j')$ is 0 for all but finitely many j'. Now suppose that $b^*_t(j)$ is positive for finitely many j. Denote this set by $\mathcal{J}_t$. In order for $b^*_{t+1}(j')$ to be positive, we need to have $\pi_{jj'} > s_{t+1}$ for some $j \in \mathcal{J}_t$. Since for each j, there are finitely many j' satisfying $\pi_{jj'} > s_{t+1}$, and since there are finitely many $j \in \mathcal{J}_t$, there can be only finitely many $\pi_{jj'} > s_{t+1}$ across *all* combinations of j and j'. Hence, b^*_{t+1} has finite support.

The backward distribution for sampling z_t given z_{t+1} and y data becomes

$$p(z_t = j \mid z_{t+1}, s, y) \propto p(z_t = j \mid s, y_1, \ldots, y_t) p(z_{t+1} \mid z_t, s) \tag{6.21}$$

$$= b^*_t(j) I(s_{t+1} > \pi_{jz_{t+1}}) \tag{6.22}$$

which is again zero for all but finitely many j.

Using this algorithm we can represent only those entries in the transition matrix π that are needed, combining the mass of all unrepresented components into one

value, $\pi_{j,new}$ for each j. When performing the forward step, each time there is a t and j such that both $b_t^*(j) > 0$,$\pi_{j,new} > s_{t+1}$, then it is possible that the chain could visit a new state, and so we perform a stick-breaking step to instantiate the new component, and the new transition probabilities to and from that component. Let J be the number of states currently instantiated, so that the current representation of β is $\beta_1, \ldots, \beta_J, \beta_{new}$. Then, to instantiate a new component, sample

$$\tilde{\beta}_{J+1} \sim \mathsf{Beta}(1, \gamma) \tag{6.23}$$

set

$$\beta_{J+1} \leftarrow \tilde{\beta}_{J+1}\beta_{new} \qquad \beta_{new} \leftarrow (1 - \tilde{\beta}_{J+1})\beta_{new} \tag{6.24}$$

and for each $j = 1, \ldots J$, sample

$$\tilde{\pi}_{j,J+1} \sim \mathsf{Beta}(\alpha\beta_{J+1}, \alpha\beta_{new}) \tag{6.25}$$

and set

$$\pi_{j,J+1} \leftarrow \tilde{\pi}_{j,J+1}\pi_{j,new} \qquad \pi_{J,new} \leftarrow (1 - \tilde{\pi}_{j,J+1})\pi_{j,new} \tag{6.26}$$

Now, if $\pi_{j,J+1} > s_{t+1}$ for some j in the support of b_t^*, then $J+1$ will be in the support of b_{t+1}^*, and we need to sample a new transition distribution from state $J+1$ according to

$$(\pi_{J+1,1}, \ldots, \pi_{J+1,J+1}, \pi_{J+1,new}) \sim \mathsf{Dirichlet}(\alpha\beta_1, \ldots, \alpha\beta_{J+1}, \alpha\beta_{new}). \tag{6.27}$$

The above process is repeated until $\pi_{j,new} < s_{t+1}$ for all j, after which we can calculate b_{t+1}^* and then move on to the next t.

6.3.2. Adapting beam sampling to the HDP-HMM-LT

In the context of the weak limit approximation to the HDP-HMM-LT, we compute the entire transition matrix, with entries proportional to $\pi_{jj'}\phi_{jj'}$. When the number

of states is bounded above by J, the beam sampling algorithm described above can be used directly, except that all J^2 transition probabilities are represented explicitly, and so there is no need for the π_{new} term and the corresponding stick-breaking step. Using beam sampling in this manner can greatly speed up the process of sampling the z_t, and thus although it does not remove the need to specify J, it allows larger J to be used without the quadratic scaling in computational complexity.

Ideally we would like to remove entirely the need to specify J. Unfortunately in the HDP-HMM-LT, the aggregated probability of transitioning to some new state is

$$\sum_{j'=J+1}^{\infty} \pi_{jj'}\phi_{jj'} \tag{6.28}$$

which cannot in general be sampled directly, since unlike in the special case when all $\phi_{jj'}$ are identical, the prior on this quantity is not a Gamma distribution. A related issue for the Markov Jump Process with Failed Transitions (MJP-FT) representation is that the auxiliary variables $\tilde{u}_t$, representing the continuous duration of interval between transitions t and $t+1$ have distributions defined in terms of the sum of transition probabilities:

$$\tilde{u}_t \mid z_t, \pi, \phi \sim \mathsf{Exponential}(\sum_{j'=1}^{\infty} \pi_{z_t,j'}\phi_{z_t,j'}) \tag{6.29}$$

It may be possible to get around this issue through the introduction of additional or modified slice variables. This is an interesting direction for future work.

6.4. An Infinite Factorial HDP-HMM-LT

One setting in which a local transition property is desireable is the case where the latent states indicate which in a set of hidden features is "active" at time t; that is, when the latent state is represented by a binary vector. A parametric example of such a model is the Factorial HMM (Ghahramani et al., 1997), nonparametric extensions of which, such as the infinite factorial hidden Markov model (Gael et al., 2009) and

the infinite factorial dynamic model (Valera et al., 2015), have been developed in recent years by making use of the Indian Buffet Process (Ghahramani and Griffiths, 2005) as a state prior. It would be conceptually straightforward to combine the IBP state prior with the similarity bias of the LT model, provided the chosen similarity function is uniformly bounded above on the space of infinite length binary vectors (for example, take $\phi(u, v)$ to be the exponentiated negative Hamming distance between u and v). Since the number of differences between two draws from the IBP is finite with probability 1, this yields a reasonable similarity metric.

Implementing and exploring this variant of the model is another direction of future research.

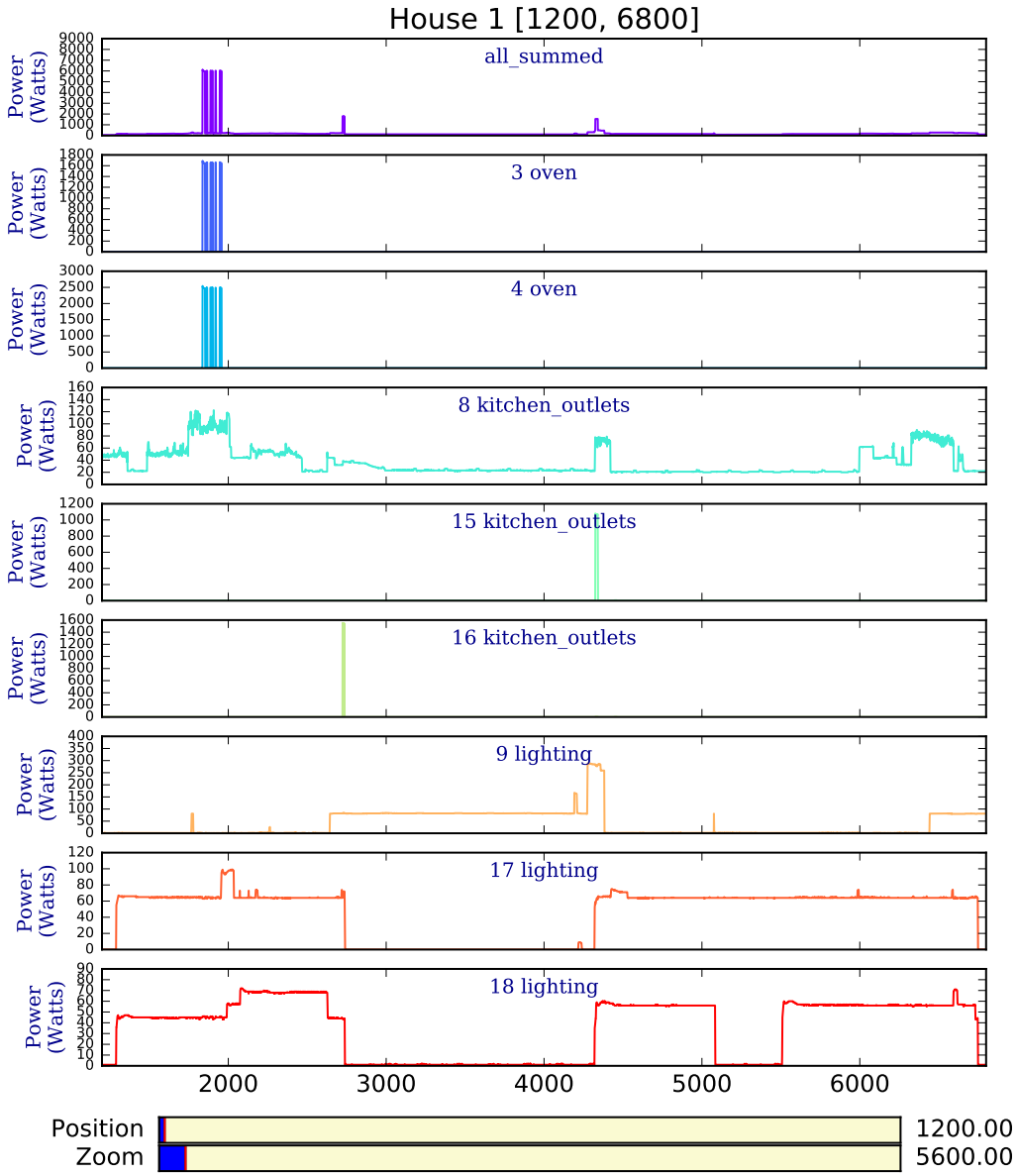


Figure 6.1: A sample data interval from the REDD dataset Kolter and Johnson (2011). The top channel contains the total measured power in watts consumed by a home during a period of approximately 24 hours. Each timestep represents a 20 second intervals, during which the amplitude recorded is a median of the amplitudes in the original higher resolution data.

A. Effect of Hyperparameters

In this appendix I examine the effect of the choices of hyperparameters on the results of two of the experiments discussed in the body of the dissertation: the synthetic cocktail party experiment in Chapter 4, and the Bach chorale experiment in Chapter 5.

A.1. Synthetic Cocktail Party Data

The results reported in Chapter 4 are based on the binary state, linear-Gaussian form of the HDP-HMM-LT, with a Laplace similarity kernel defined over Hamming distances between binary state vectors. All hyperparameters of the HDP, as well as the scale parameter of the similarity kernel, are sampled during inference; however, the hyperpriors on the two HDP concentration parameters, α and γ, and on the variance, σ^2 of the Gaussian noise, need to be set by hand. In this section I explore the effect of different choices of hyperparameters for these priors.

The HDP-HMM and the HDP-HMM-LT as used in their binary state, linear-Gaussian forms, are defined with the following generative model:

$$\begin{aligned} \beta &\sim \mathsf{GEM}(\gamma) \\ \pi_{jj'} &\overset{i.i.d}{\sim} \mathsf{Gamma}(\alpha\beta_{j'}, \phi_{jj'}^{-1}) \quad j' = 1, 2, \ldots \\ z_t \,|\, z_{t-1} &\sim \mathsf{Discrete}(\pi_{z_t,\cdot}) \\ y_{tk} \,|\, z_t, \theta, w_k &\overset{ind.}{\sim} \mathcal{N}(w_k^{\mathsf{T}}\theta_{z_t}, h_k^{-1}) \end{aligned}$$

where in the non-LT HDP-HMM case all $\phi_{jj'}$ are identically 1, while in the LT case, we have

$$\phi_{jj'} := \exp(-\lambda\Delta_{j,j'})$$

Expt.	a_γ	b_γ	a_α	b_α	a_h	b_h	b_λ
1	**0.01**	**0.01**	0.1	0.1	0.1	0.1	0.1
2	**0.01**	**5.0**	0.1	0.1	0.1	0.1	0.1
3	**5.0**	**0.01**	0.1	0.1	0.1	0.1	0.1
4	**5.0**	**5.0**	0.1	0.1	0.1	0.1	0.1
5	0.1	0.1	**0.01**	**0.01**	0.1	0.1	0.1
6	0.1	0.1	**0.01**	**5.0**	0.1	0.1	0.1
7	0.1	0.1	**5.0**	**0.01**	0.1	0.1	0.1
8	0.1	0.1	**5.0**	**5.0**	0.1	0.1	0.1
9	0.1	0.1	0.1	0.1	**0.01**	**0.01**	0.1
10	0.1	0.1	0.1	0.1	**0.01**	**5.0**	0.1
11	0.1	0.1	0.1	0.1	**5.0**	**0.01**	0.1
12	0.1	0.1	0.1	0.1	**5.0**	**5.0**	0.1

Table A.1: Hyperprior parameter values explored in the synthetic cocktail party setting

where $\Delta_{jj'} = \sum_d |\theta_{jd} - \theta_{jd'}|$ is the Hamming distance between θ_j and $\theta_{j'}$.

In the case of the Binary Factorial HMM, the binary transition matrices for each were also constructed using independent HDP priors with concentration parameters γ and α.

We have the following top-level priors:

$$\gamma \sim \mathsf{Gamma}(a_\gamma, b_\gamma)$$

$$\alpha \sim \mathsf{Gamma}(a_\alpha, b_\alpha)$$

$$h \sim \mathsf{Gamma}(a_h, b_h)$$

$$\lambda \sim \mathsf{Exponential}(b_\lambda)$$

where all distributions are parameterized in terms of shape a and rate b.

The reference value for all shape and rate parameters is 0.1. The parametric variants explored in the results reported here are listed in Table A.1.

For the most part the hyperparameters make little difference to the pattern of results, with the exception that when the b_α or b_γ parameters are too high, and hence the prior is holding α and γ down at smaller values. The prior mean of a Gamma

distribution is a/b and the prior variance is a/b^2, so when a is 0.01 and b is 5, the mean is 0.002 and the standard deviation is 0.02, which means that even 1.0 is nearly 50 standard deviations above the mean; when $a = b = 5.0$, the mean is 1 and the standard deviation is less than 0.5. Under these conditions, all of the HDP models suffer: in the case of γ being too small, they concentrate their mass too tightly in a few states overall, and in the case of α being too small, the transition matrices become too close to deterministic (with too few states in each row having non-negligible probability mass). However, under all less restrictive priors, the pattern of results reported in Chapter 4 holds.

A.2. Bach Chorale Data

The models used in the music experiments in Chapter 5 employ separate emission and similarity spaces, and hence, conditioned on the state sequence, the observations do not inform the locations, ℓ_j. As a result there is no extrinsic information about the scale of the similarity kernel, since performing a rescaling of the locations ℓ_j together with an inverse scaling of the kernel parameter λ results in an equivalent likelihood. Therefore, rather than letting λ be inferred as it is in the cocktail party experiments, its value was fixed, whereas the locations are given a multivariate $\mathcal{N}(0, \mathbf{I})$ prior (in all experiments reported here the dimension of the similarity space is 2) and are sampled using Hamiltonian Monte Carlo.

Although sampling state locations in principle allows similarities to be learned, the choice of scale parameter λ relative to the prior on state locations is consequential in practice. In Figs. A.25-A.27, training and test set log likelihoods are shown on the Bach chorale data for four models — HDP-HMM, Sticky HDP-HMM, HDP-HMM-LT, and Sticky HDP-HMM-LT — with λ values of 0.01, 1.0 and 5.0, respectively. In Fig. A.28, the effect of λ on the log likelihood is shown for a spectrum of models ranging from the original HDP-HMM ($\lambda = 0$) to an HDP-HMM-LT with $\lambda = 10$.

The pattern reported in Chapter 5 can be seen clearly in these results: with very small λ (0.01), the LT models are nearly identical to the non-LT models. As λ increases up to 5, the gap increases, with the LT models achieving lower training log likelihoods and higher test log likelihoods, reflecting the tendency of the non-LT models' less restrictive prior on transition matrices to overfit by employing as many states as possible. When λ gets as large as 10, however, performance degrades, with both training and test log likelihoods decreasing. This likely reflects a too-strong bias toward self-transitions, resulting in the use of too few states and underfitting the data.

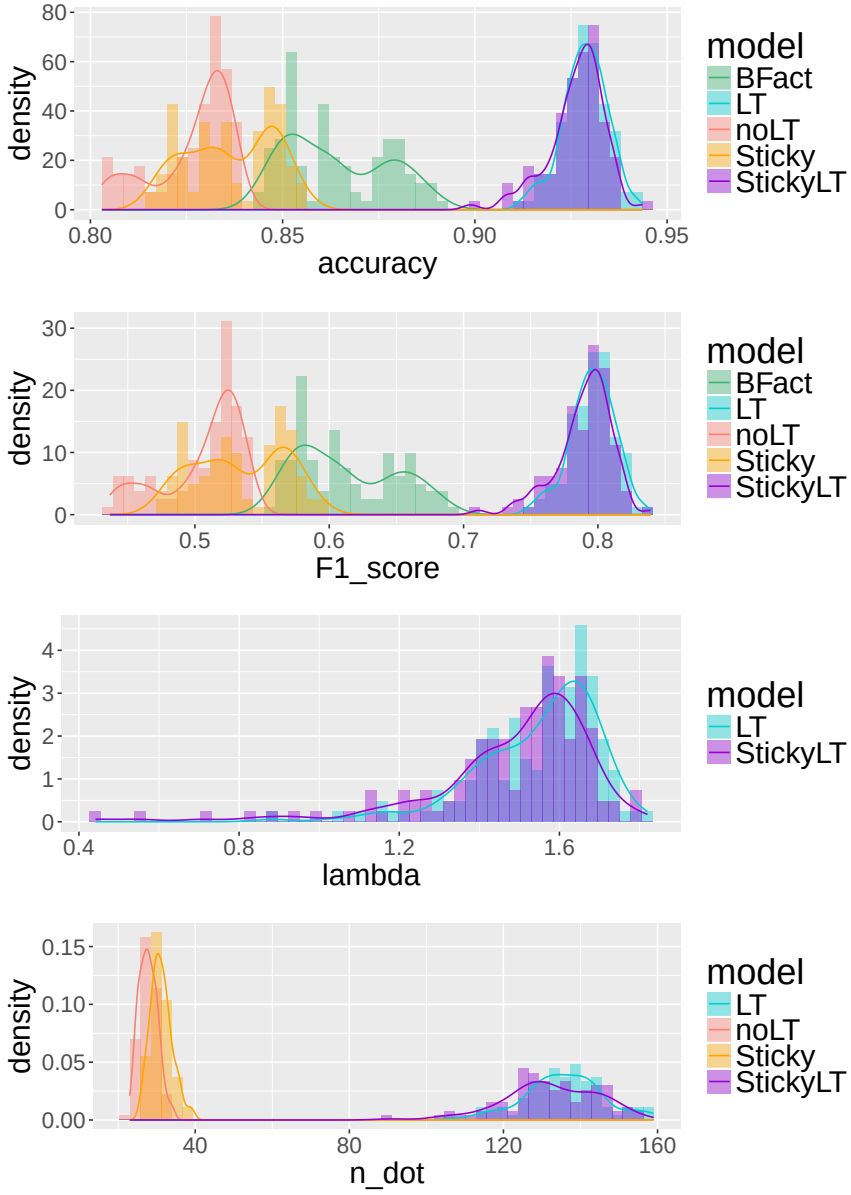


Figure A.1: Metrics for run 1: $\gamma \sim \mathsf{Gamma}(0.01, 0.01)$

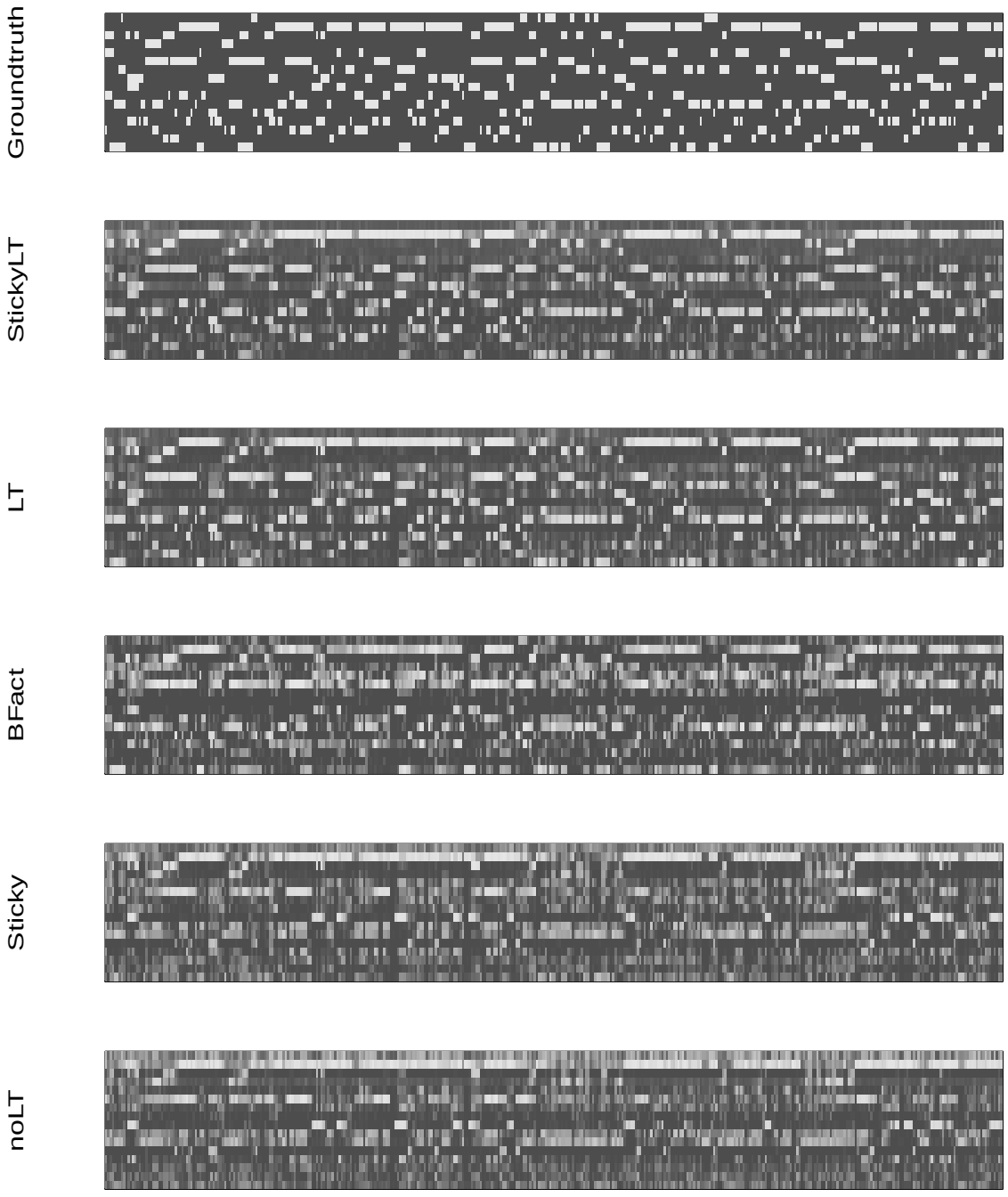


Figure A.2: Binary speaker matrices for run 1: $\gamma \sim \mathsf{Gamma}(0.01, 0.01)$

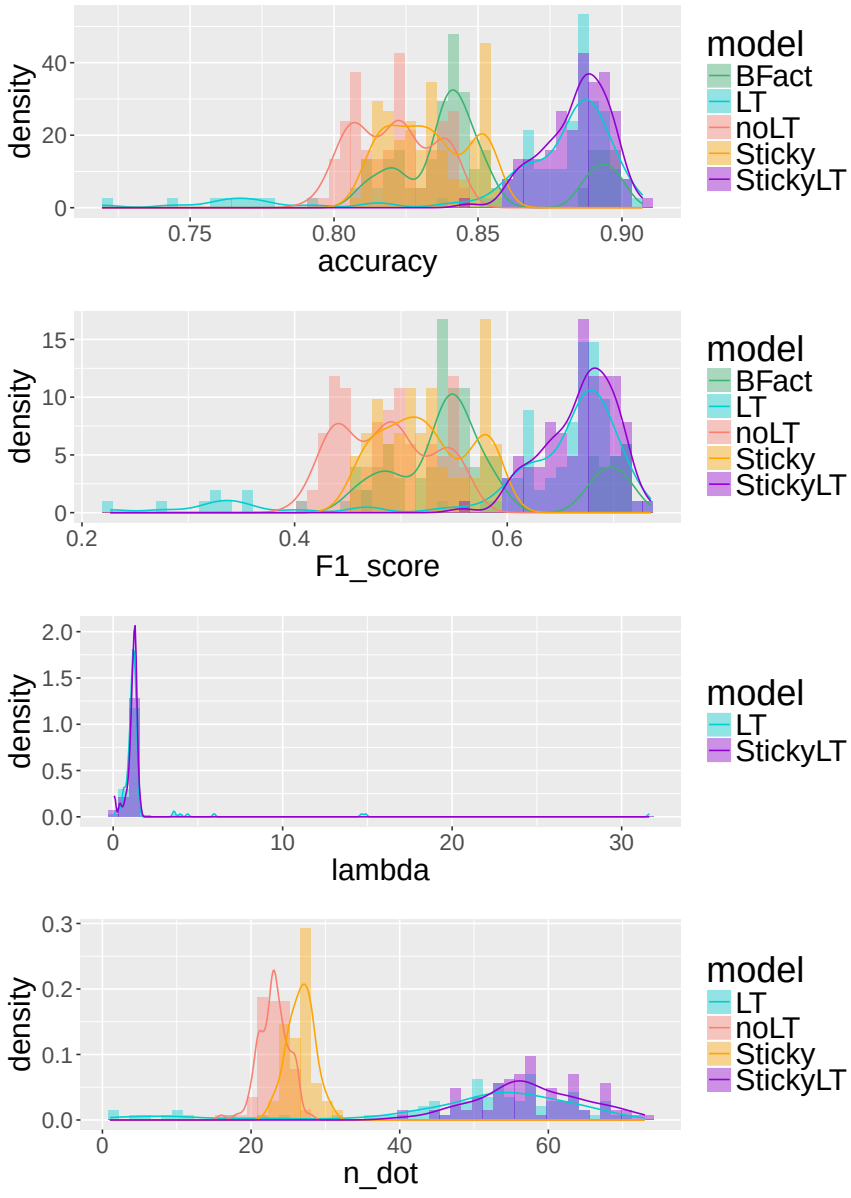


Figure A.3: Metrics for run 2: $\gamma \sim \mathsf{Gamma}(0.01, 5)$

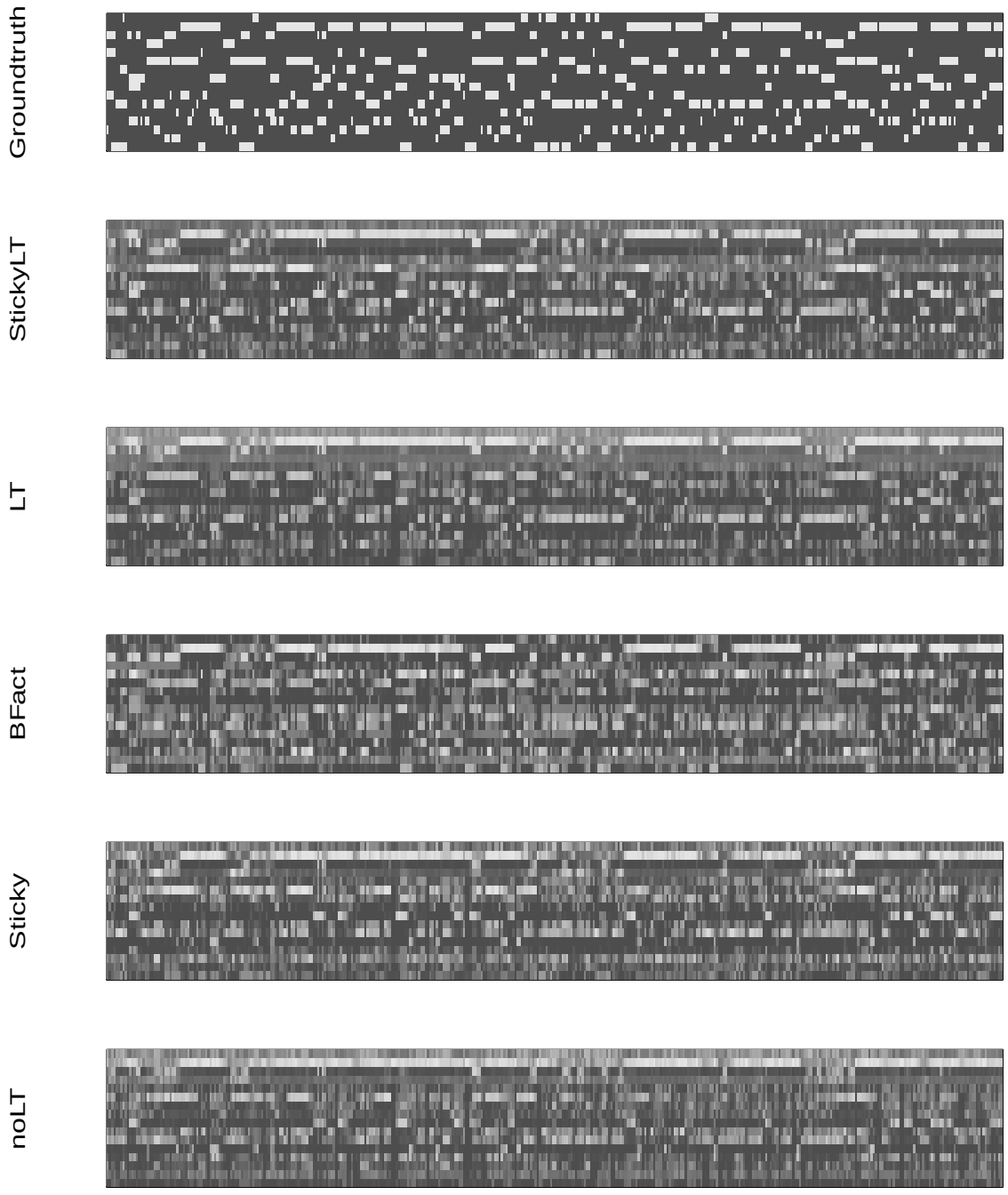


Figure A.4: Binary speaker matrices for run 2: $\gamma \sim \mathsf{Gamma}(0.01, 5)$

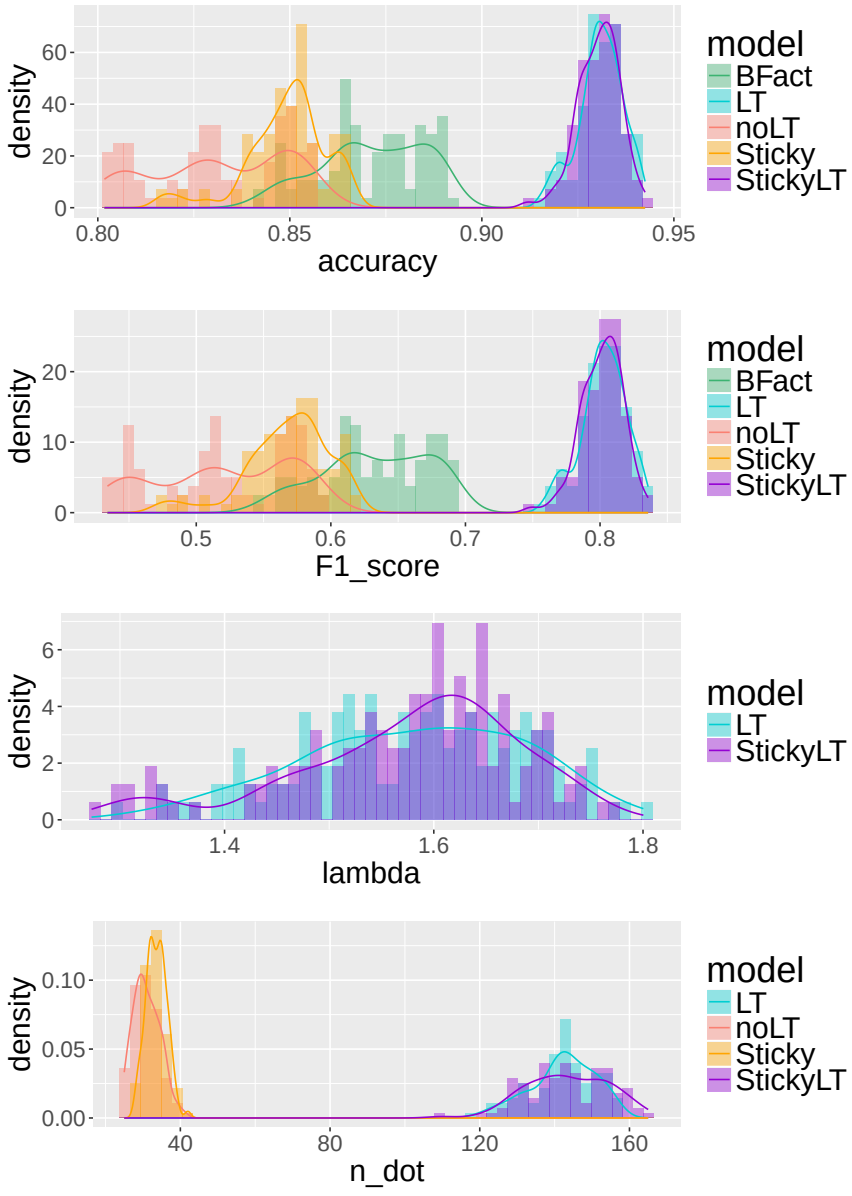


Figure A.5: Metrics for run 3: $\gamma \sim \mathsf{Gamma}(5, 0.01)$

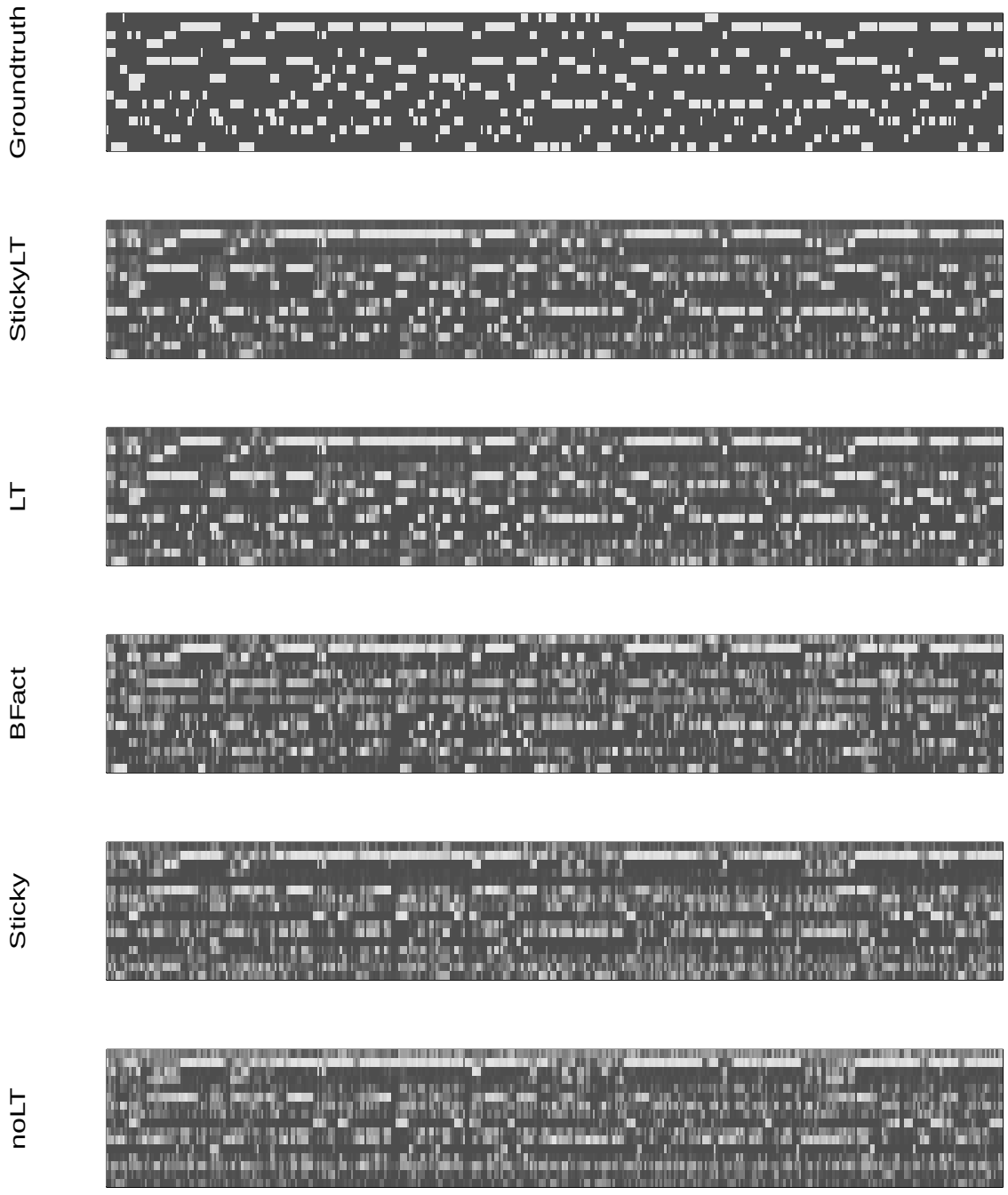


Figure A.6: Binary speaker matrices for run 3: $\gamma \sim \mathsf{Gamma}(5, 0.01)$

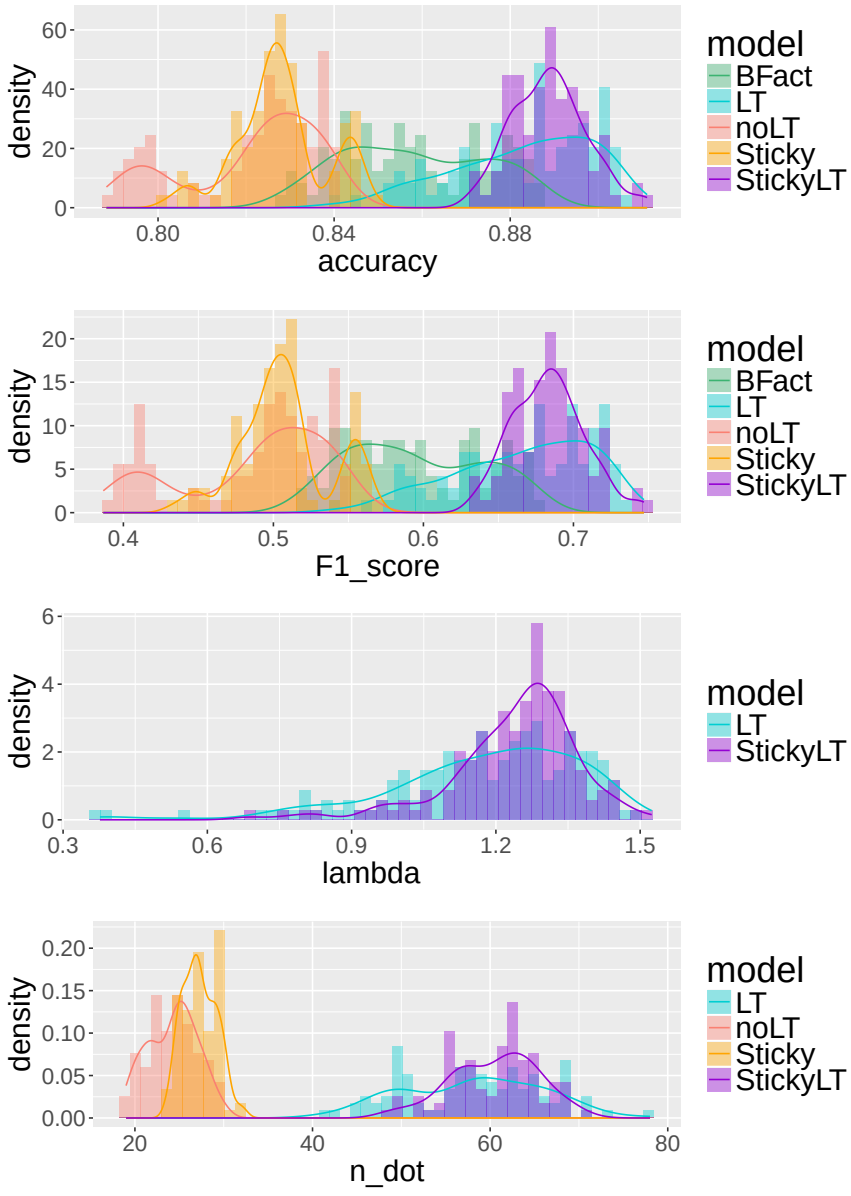


Figure A.7: Metrics for run 4: $\gamma \sim \mathsf{Gamma}(5, 5)$

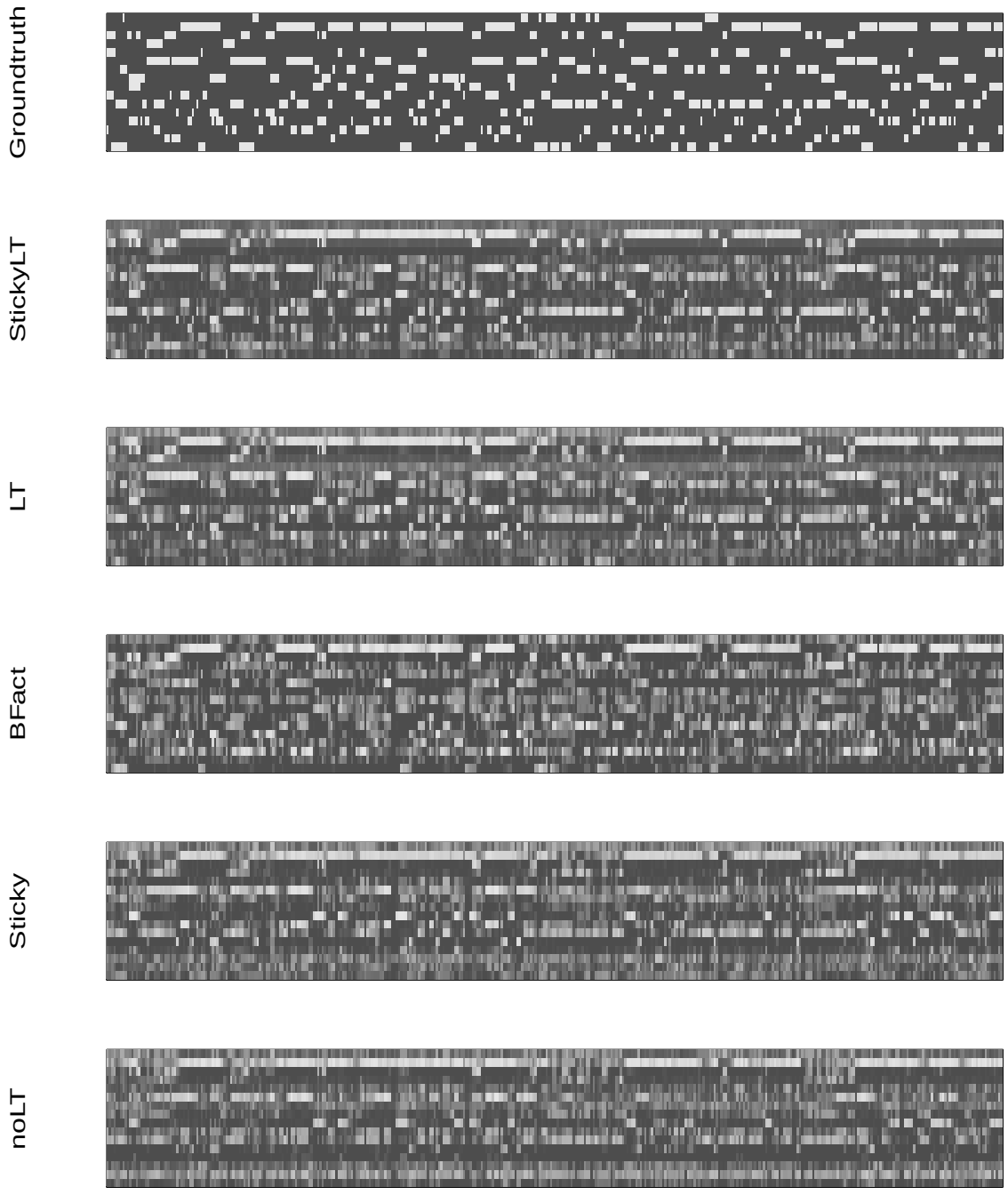


Figure A.8: Binary speaker matrices for run 4: $\gamma \sim \mathsf{Gamma}(5, 5)$

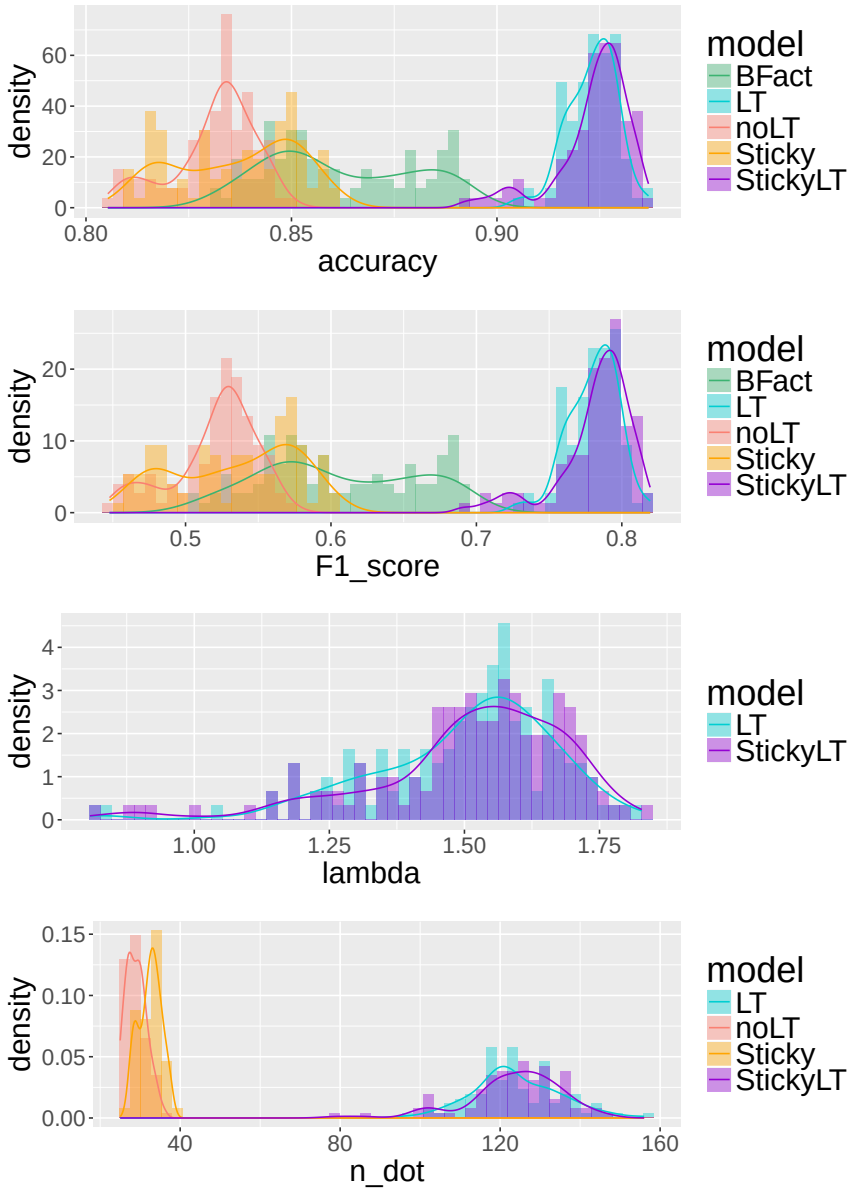


Figure A.9: Metrics for run 5: $\alpha \sim \mathsf{Gamma}(0.01, 0.01)$

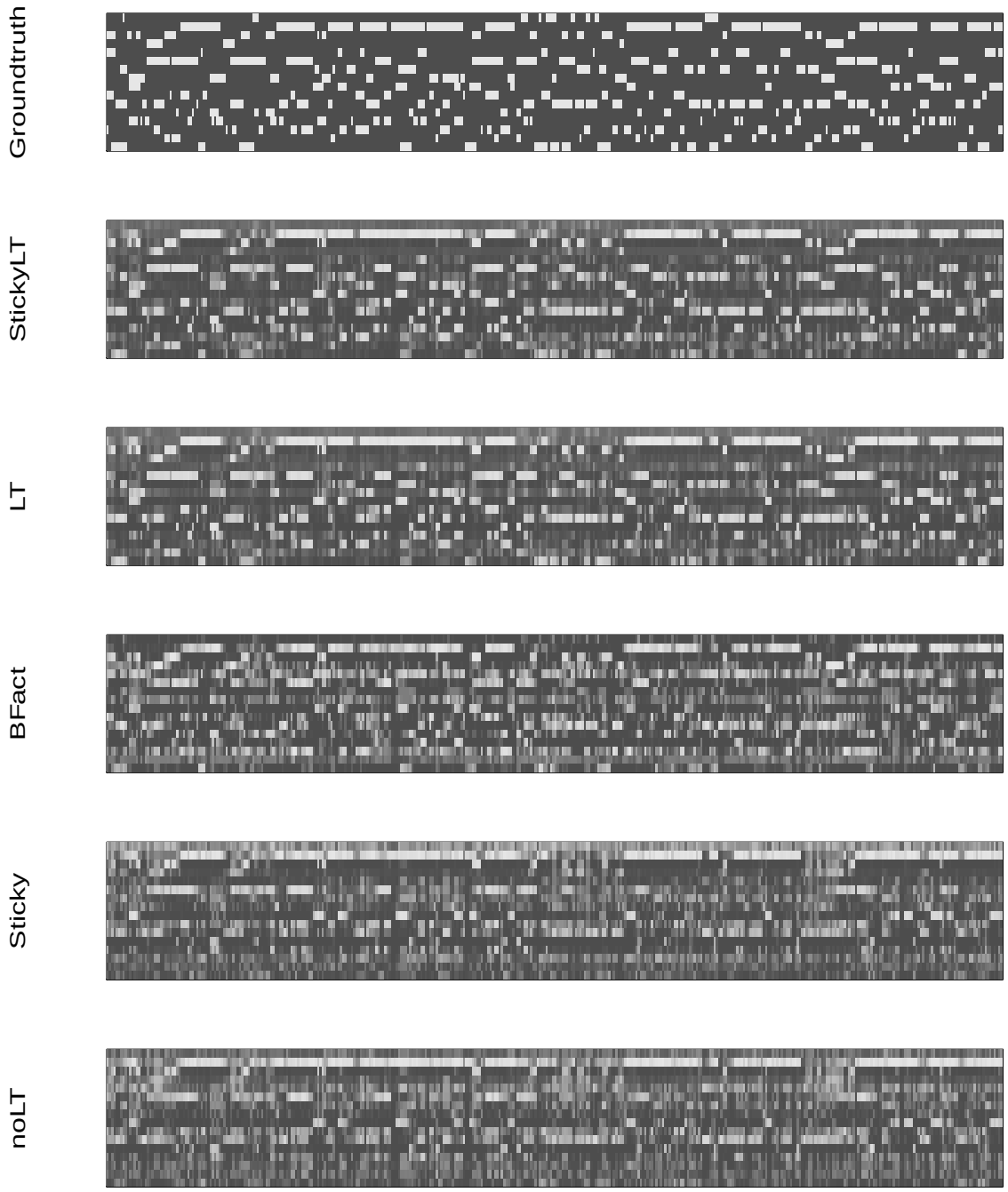


Figure A.10: Binary speaker matrices for run 5: $\alpha \sim \mathsf{Gamma}(0.01, 0.01)$

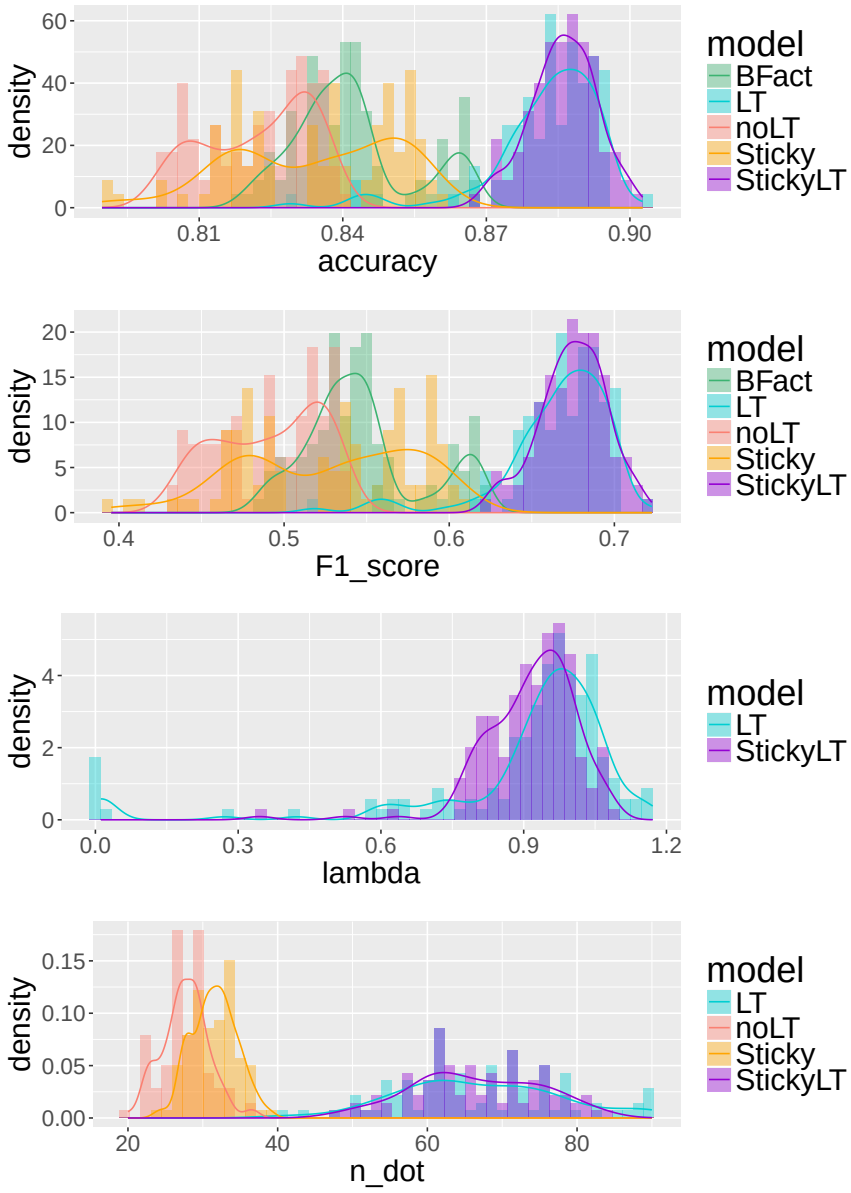


Figure A.11: Metrics for run 6: $\alpha \sim \mathsf{Gamma}(0.01, 5)$

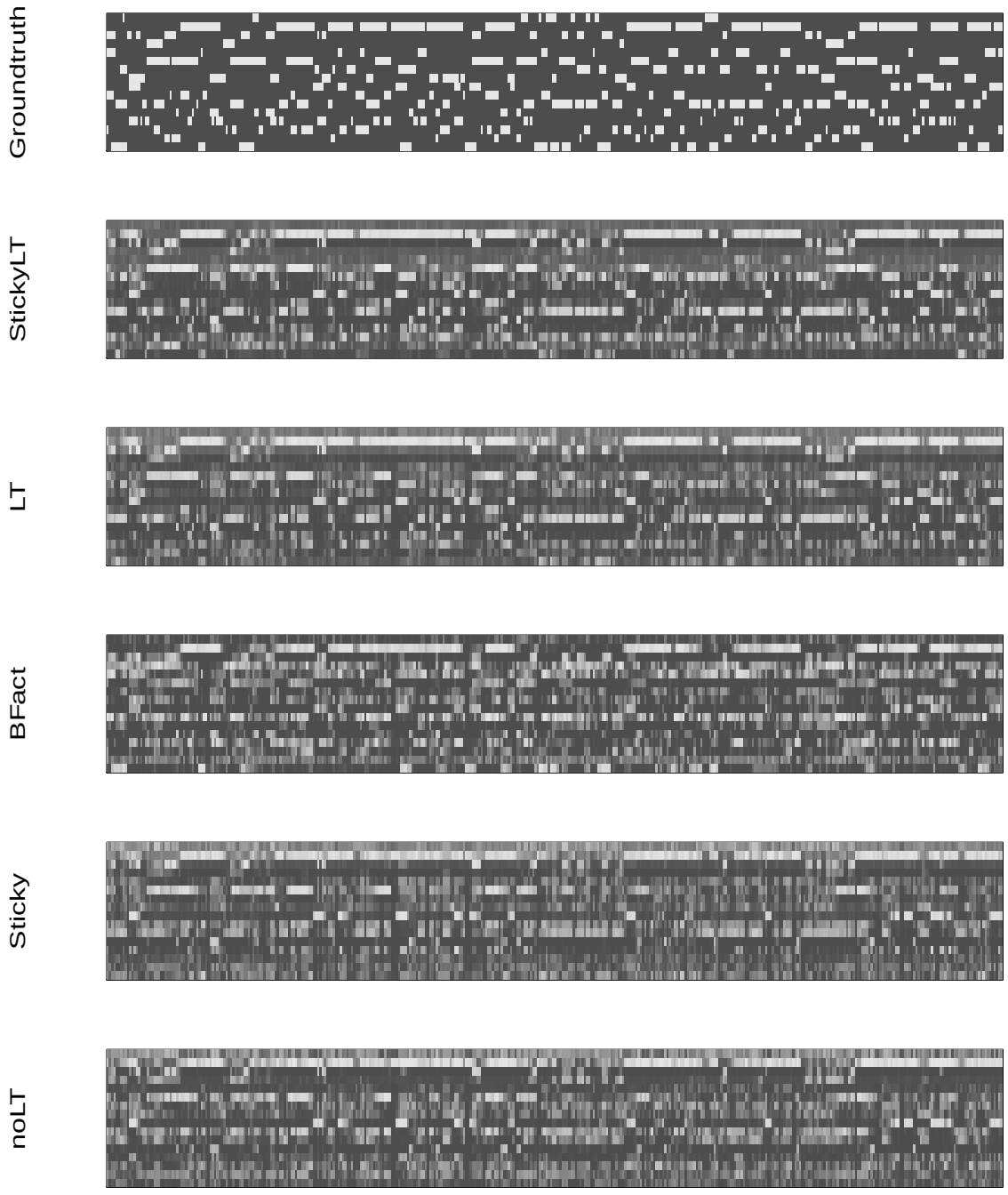


Figure A.12: Binary speaker matrices for run 6: $\alpha \sim \mathsf{Gamma}(0.01, 5)$

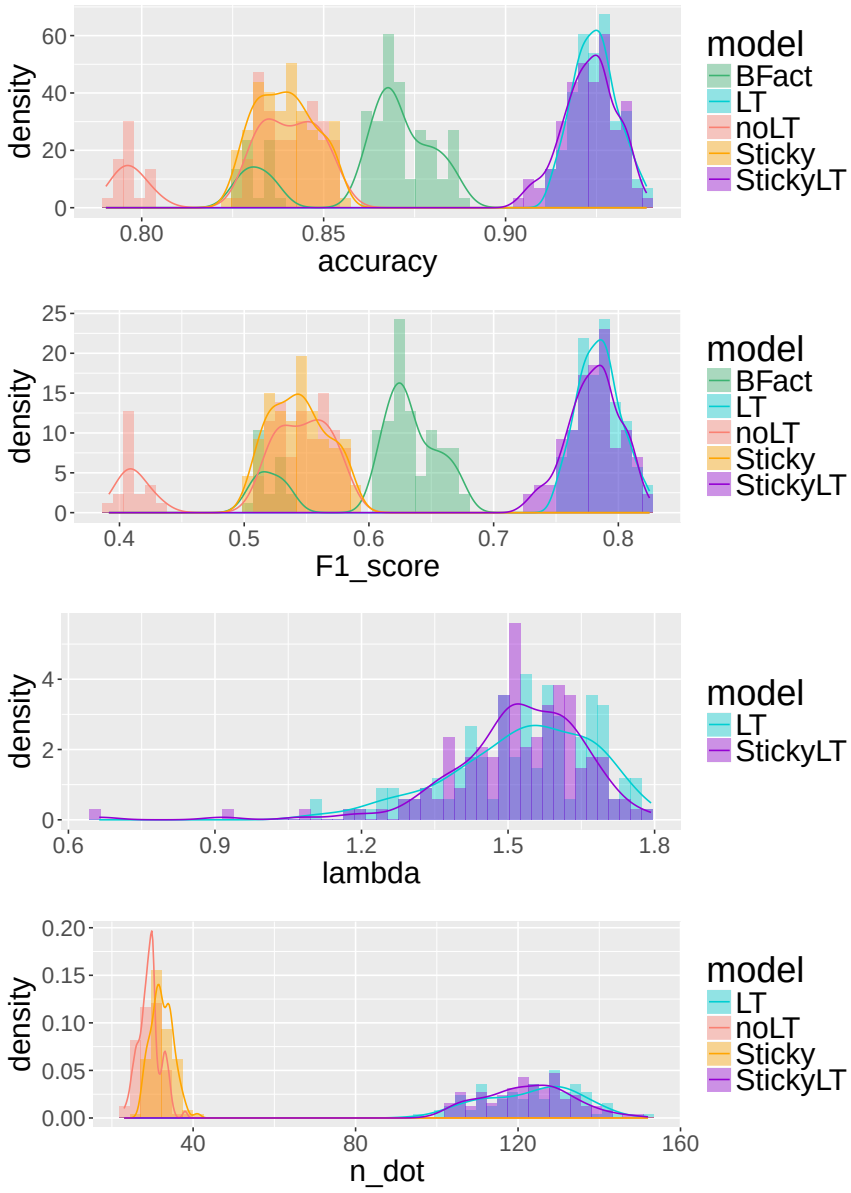


Figure A.13: Metrics for run 7: $\alpha \sim \mathsf{Gamma}(5, 0.01)$

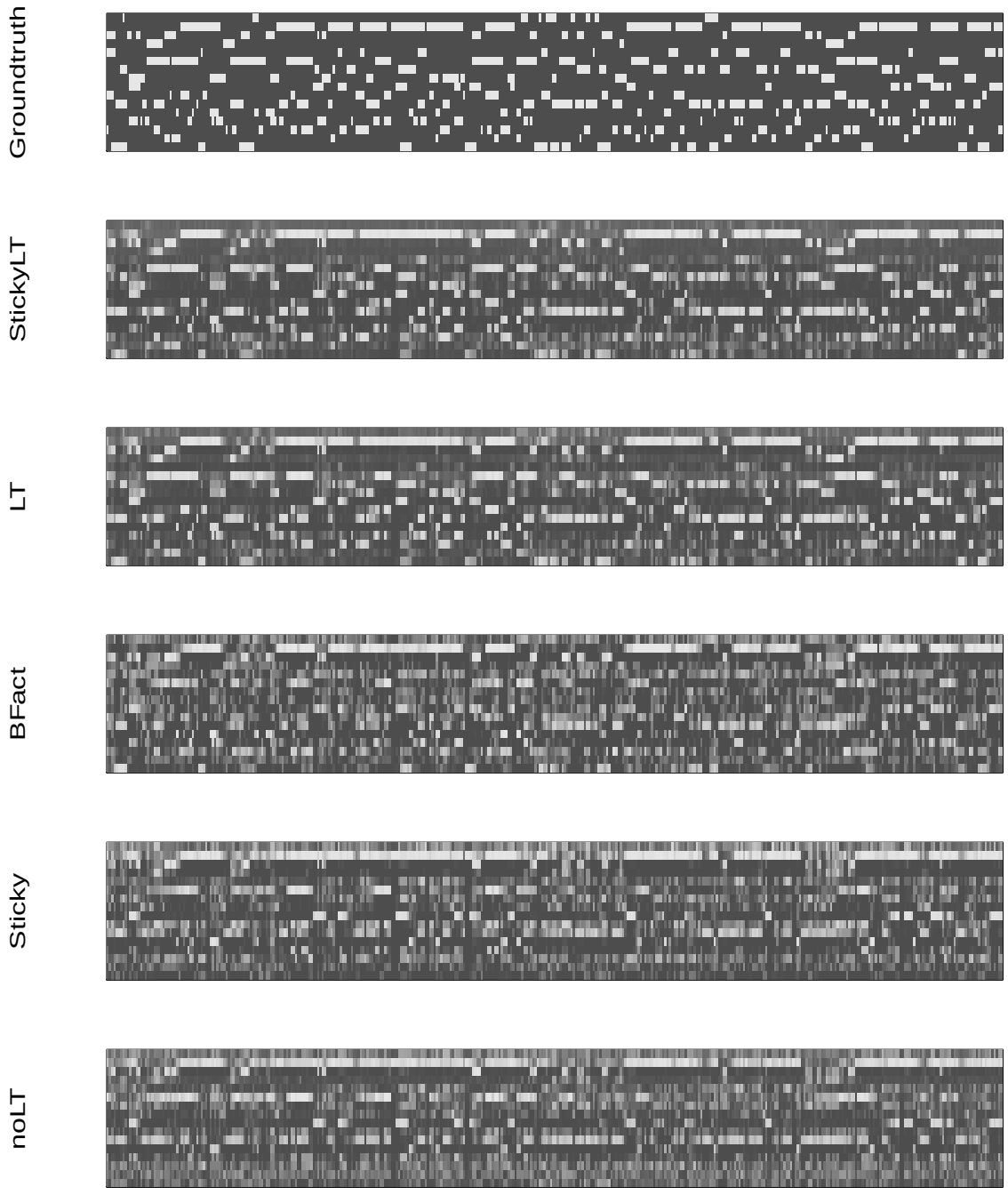


Figure A.14: Binary speaker matrices for run 7: $\alpha \sim \mathsf{Gamma}(5, 0.01)$

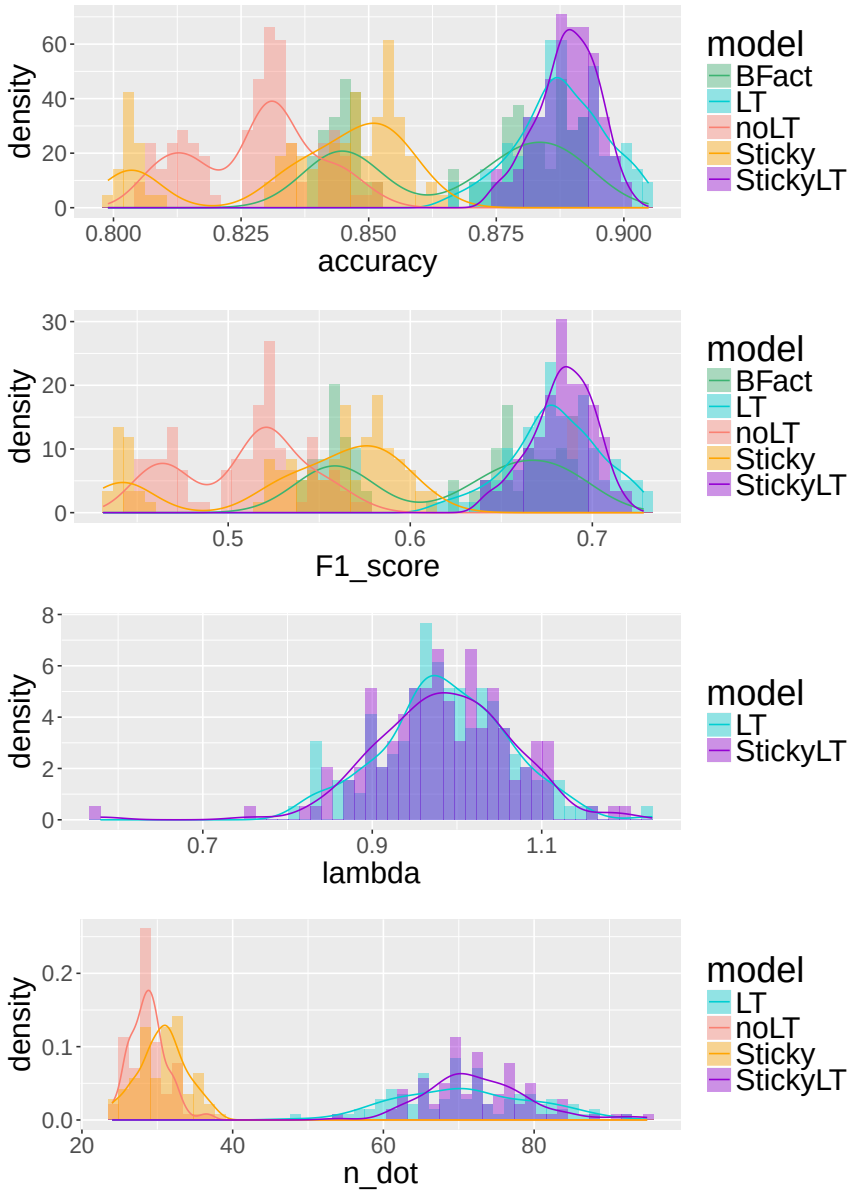


Figure A.15: Metrics for run 8: $\alpha \sim \mathsf{Gamma}(5, 5)$

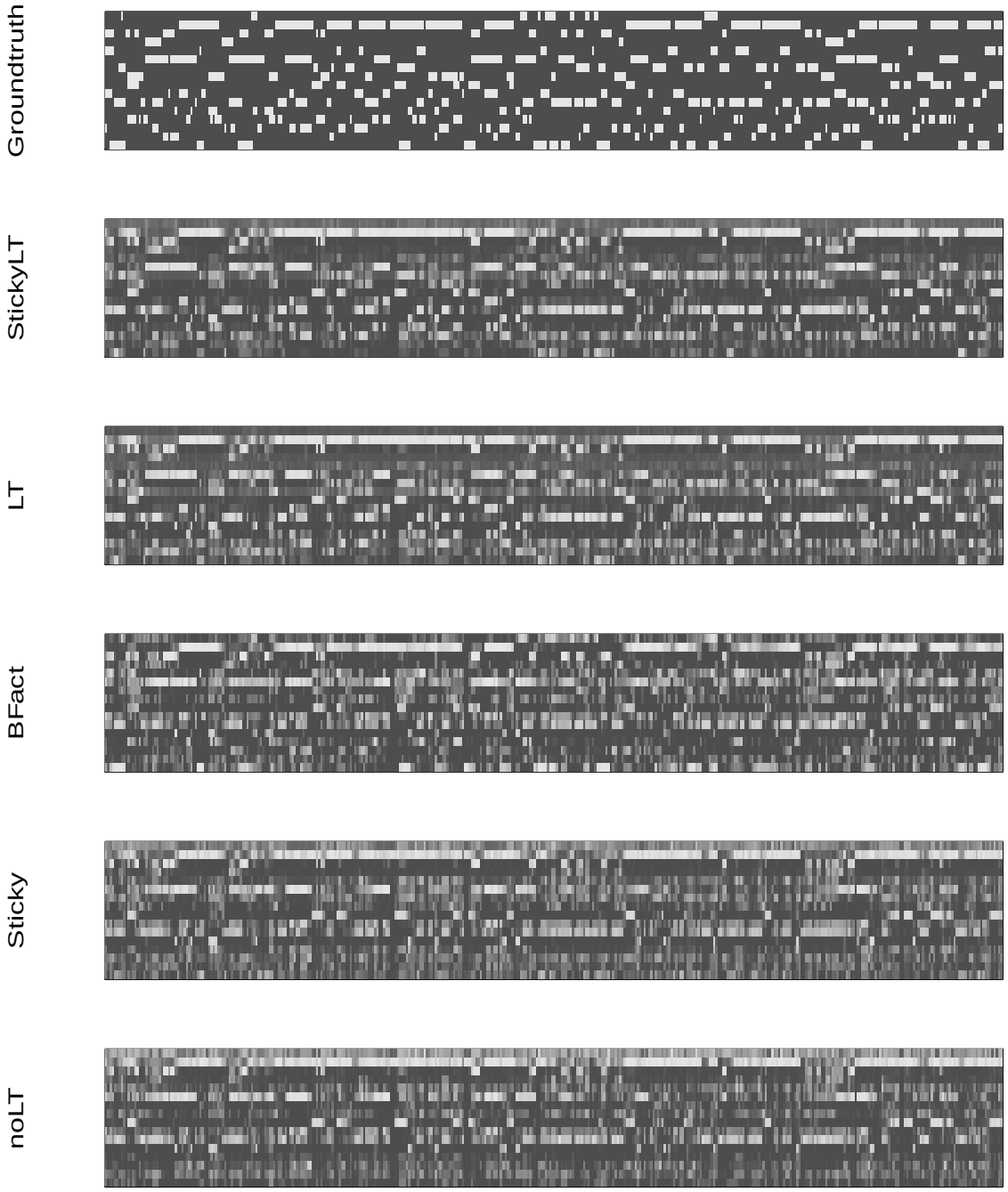


Figure A.16: Binary speaker matrices for run 8: $\alpha \sim \mathsf{Gamma}(5, 5)$

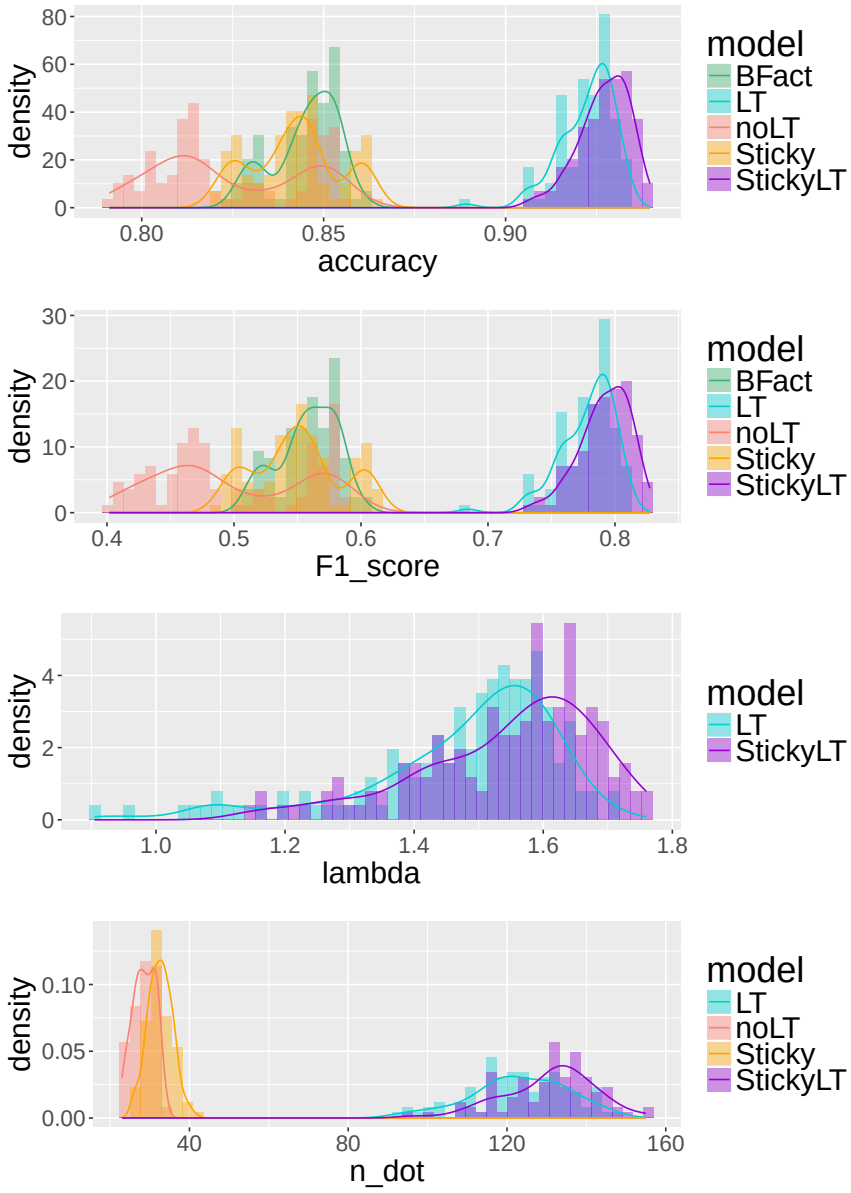


Figure A.17: Metrics for run 9: $h \sim \mathsf{Gamma}(0.01, 0.01)$

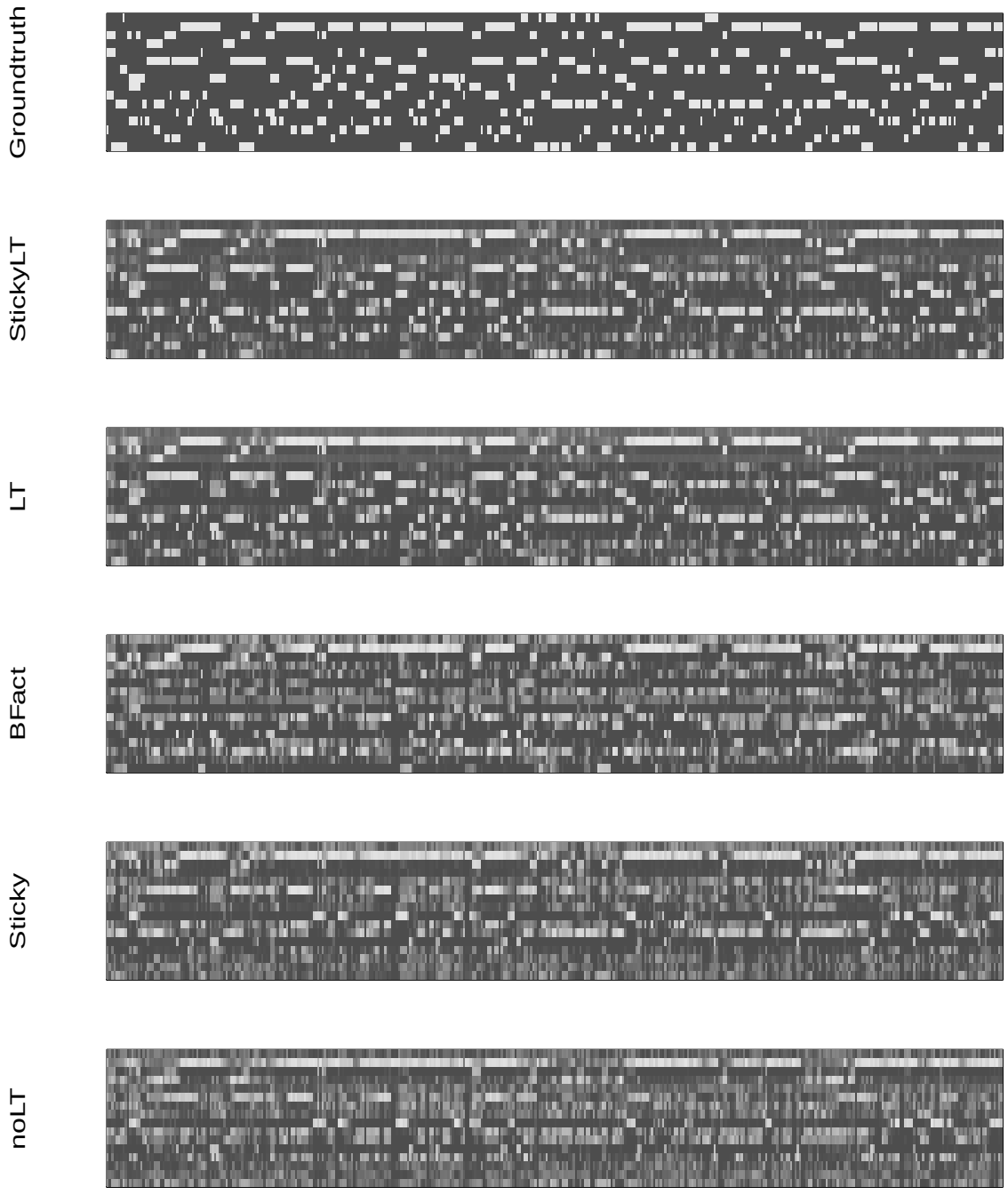


Figure A.18: Binary speaker matrices for run 9: $h \sim \mathsf{Gamma}(0.01, 0.01)$

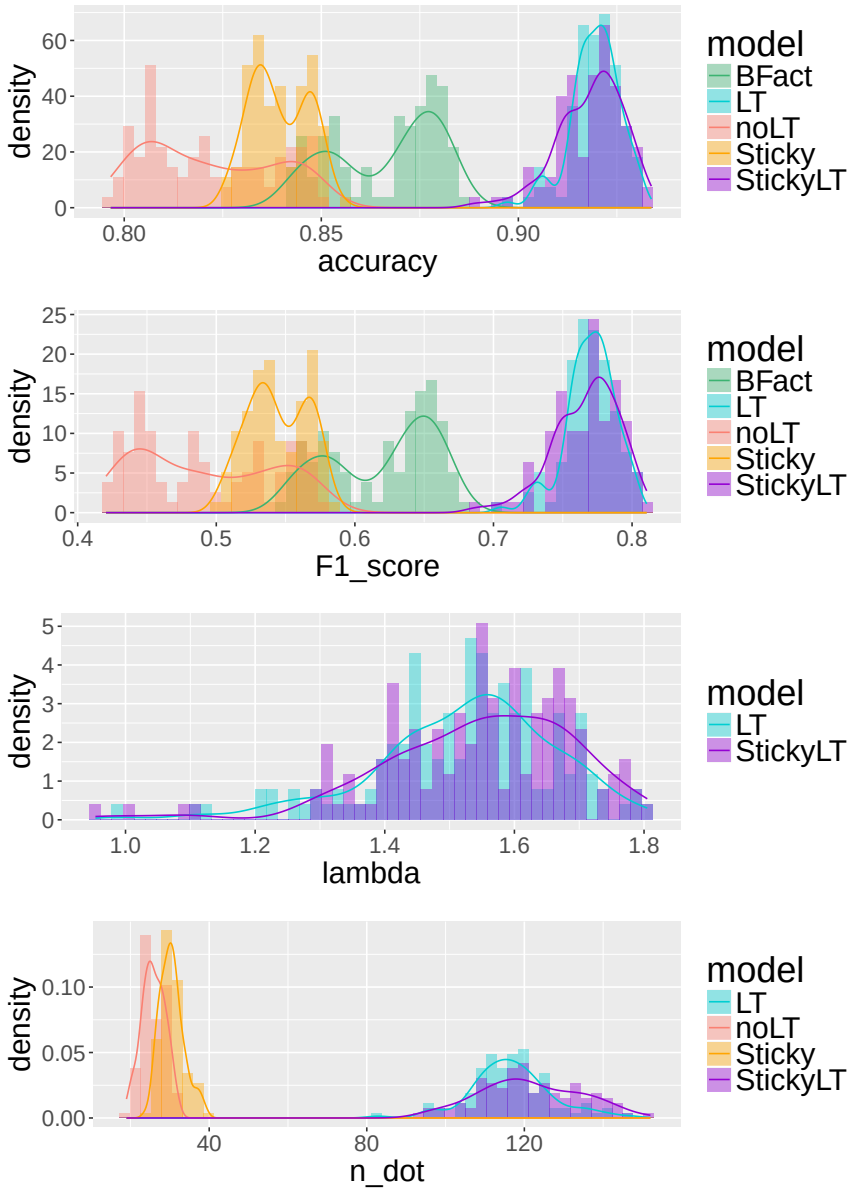


Figure A.19: Metrics for run 10: $h \sim \mathsf{Gamma}(0.01, 5)$

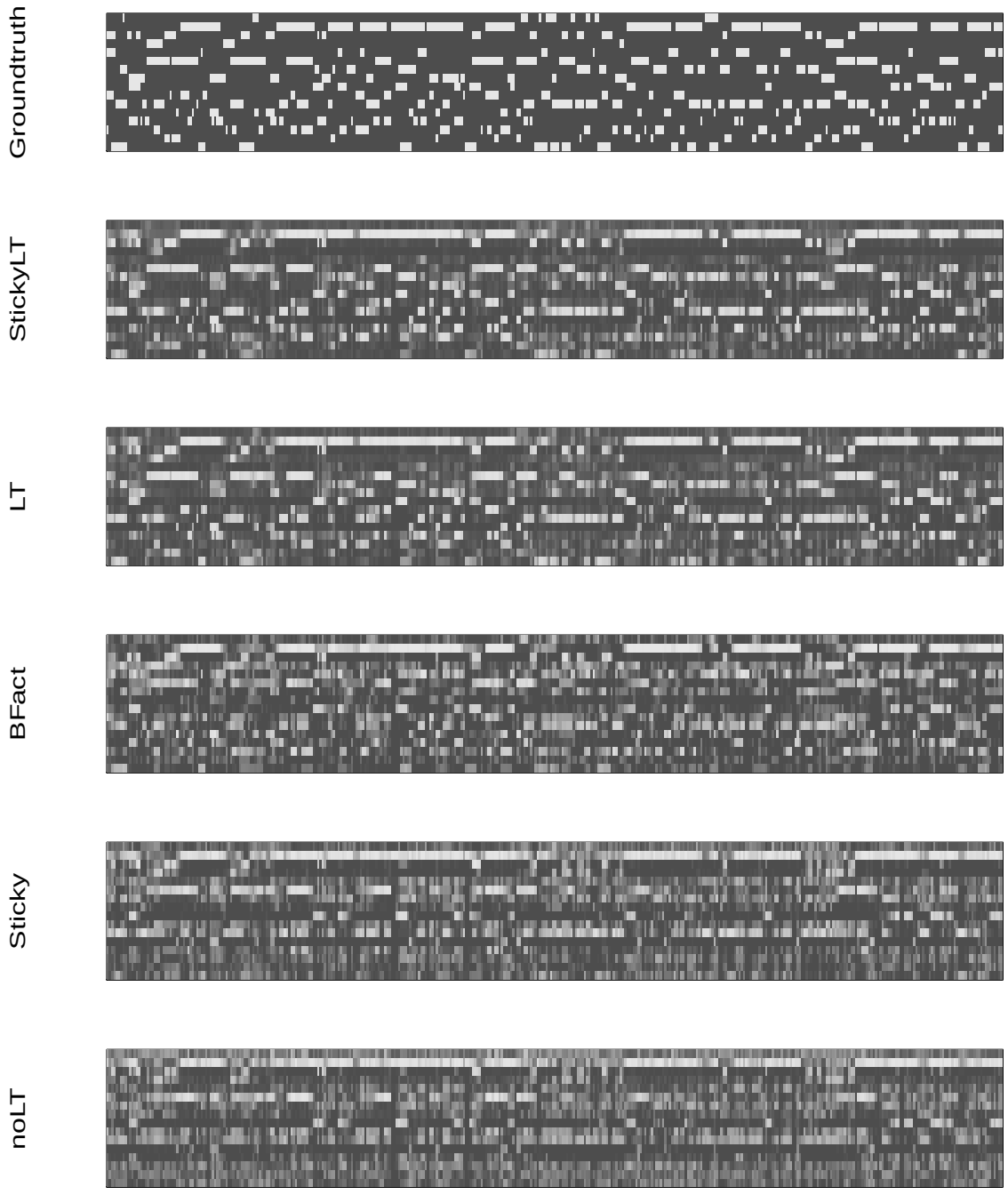


Figure A.20: Binary speaker matrices for run 10: $h \sim \mathsf{Gamma}(0.01, 5)$

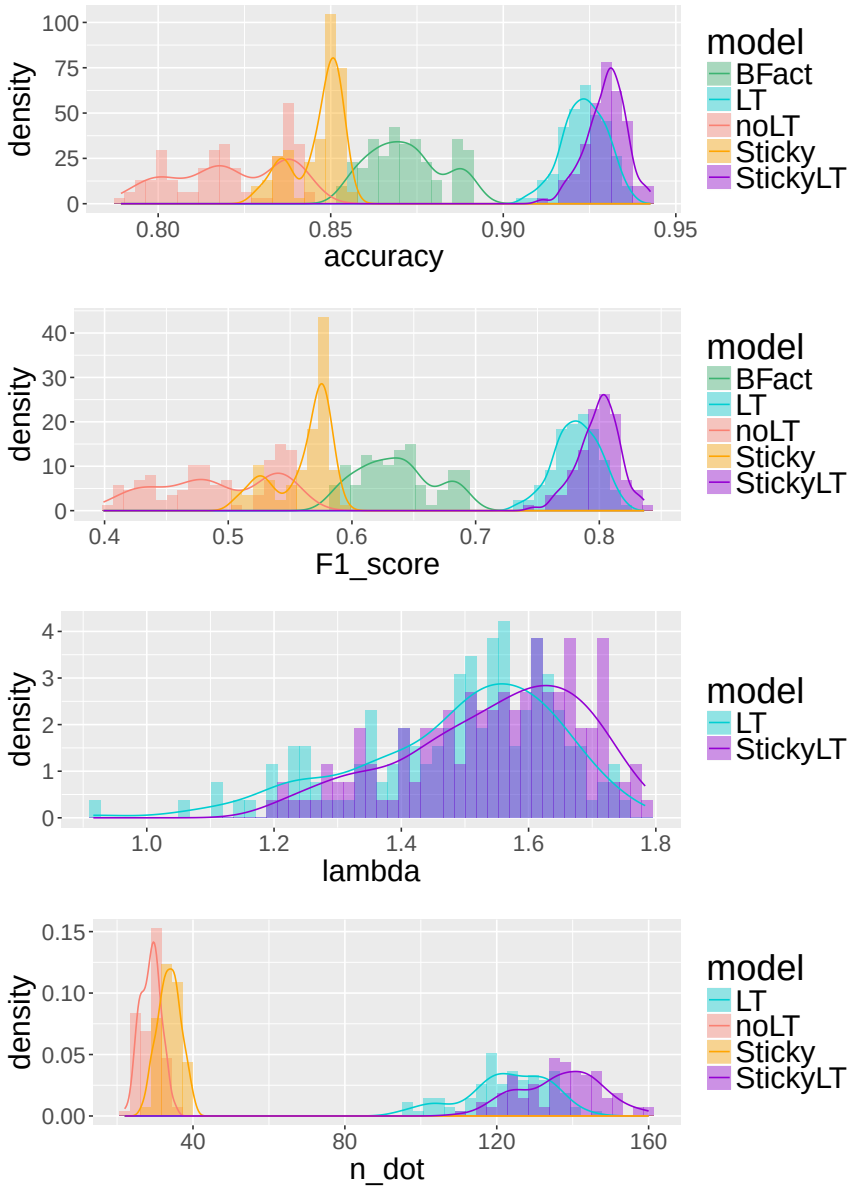


Figure A.21: Metrics for run 11: $h \sim \mathsf{Gamma}(5, 0.01)$

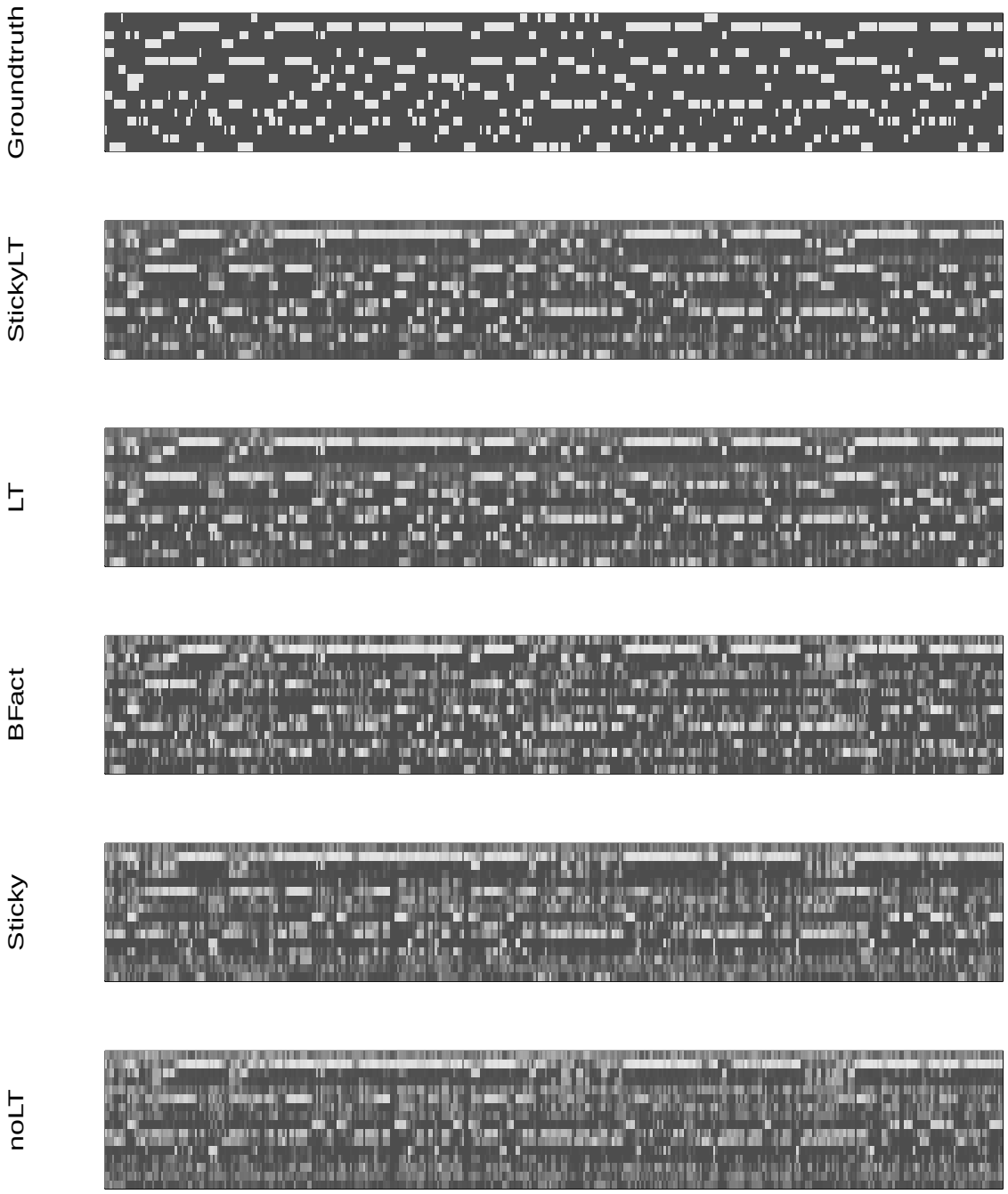


Figure A.22: Binary speaker matrices for run 11: $h \sim \mathsf{Gamma}(5, 0.01)$

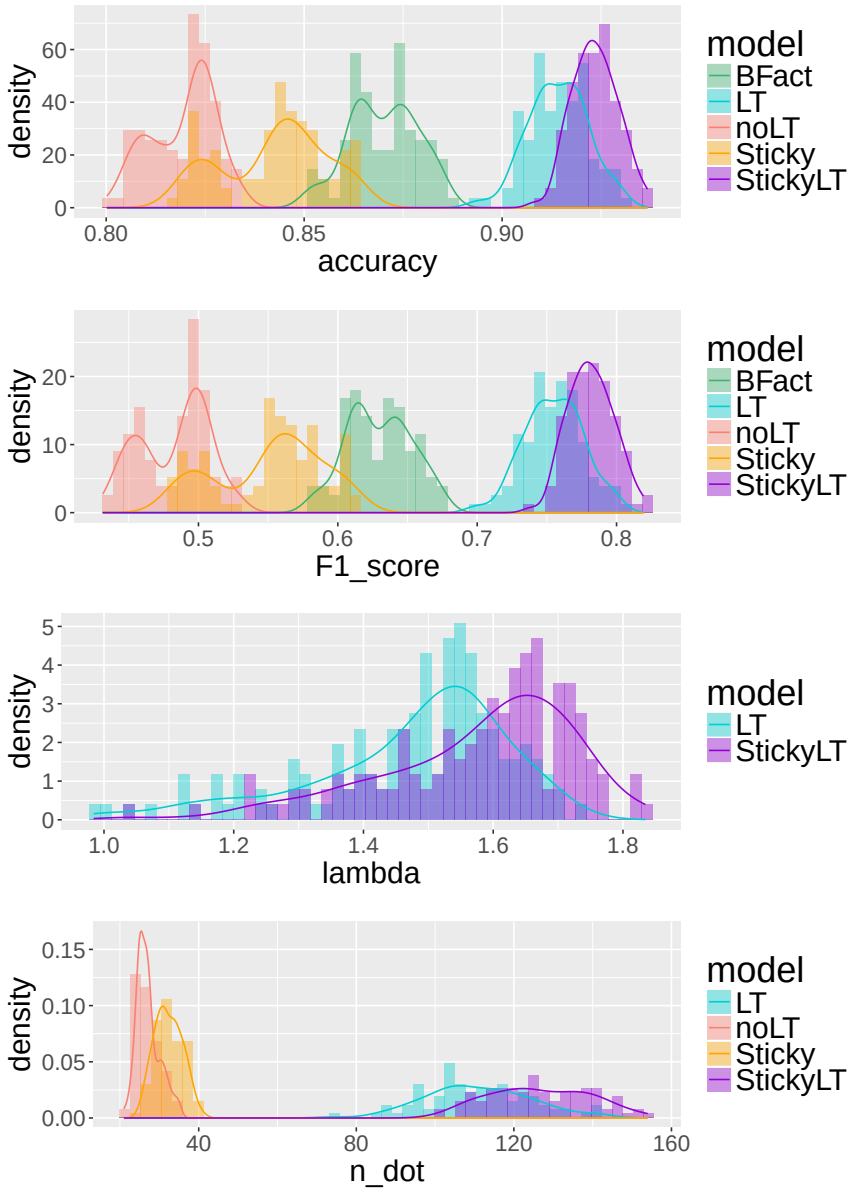


Figure A.23: Metrics for run 12: $h \sim \mathsf{Gamma}(5, 5)$

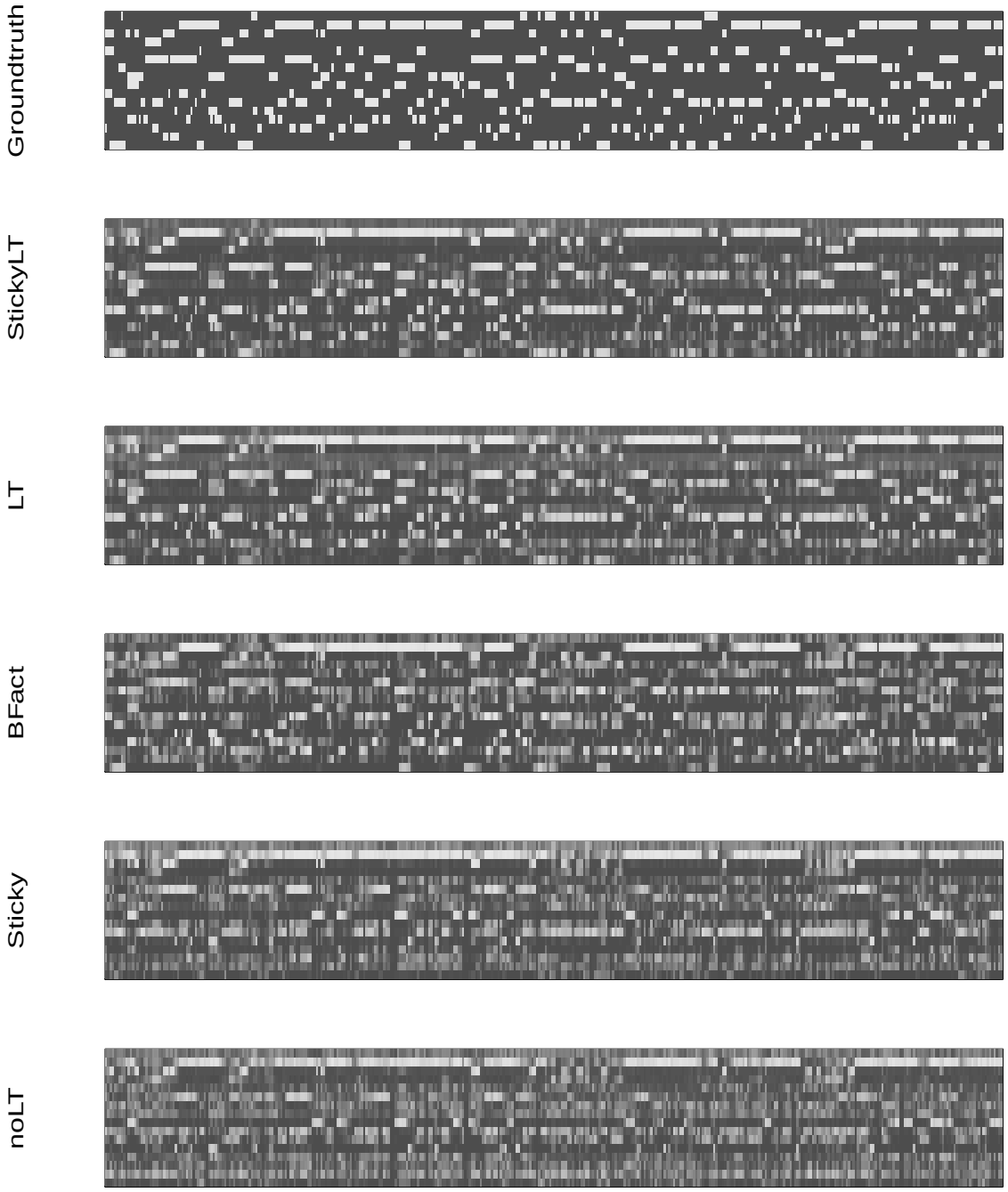


Figure A.24: Binary speaker matrices for run 12: $h \sim \mathsf{Gamma}(5, 5)$

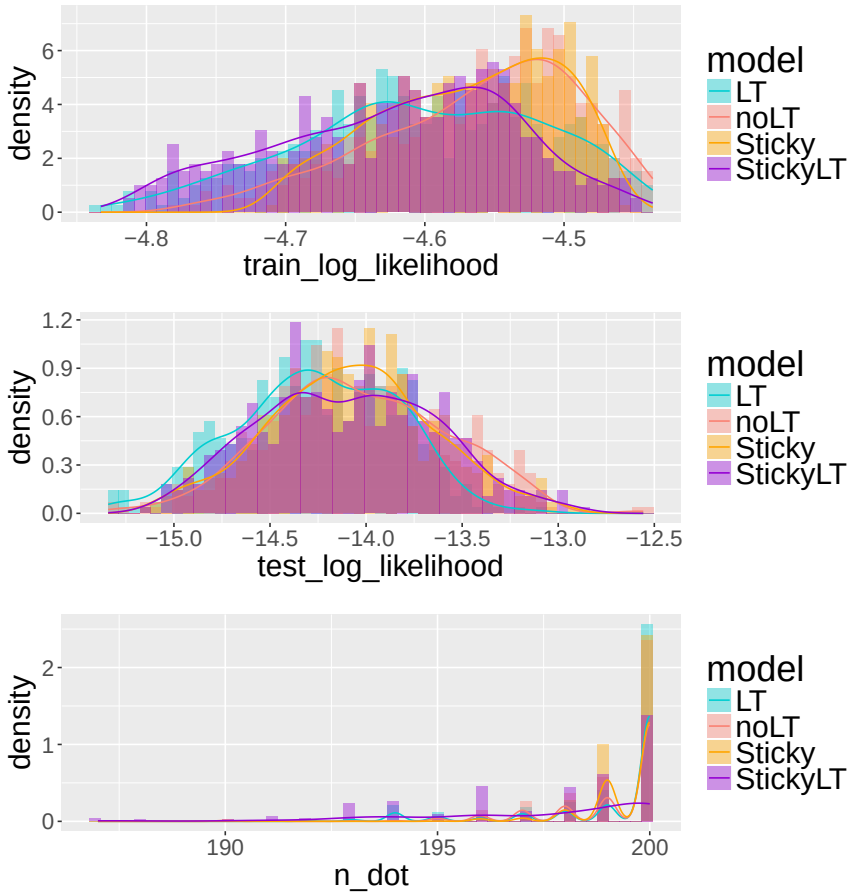


Figure A.25: Top and Middle: Training set and test set log marginal likelihoods for Bach chorale data on the four HDP-based models: HDP-HMM-LT, HDP-HMM, Sticky HMM, and Sticky HDP-HMM-LT, where the two LT models set $\lambda = 0.01$. Bottom: Number of latent states occupied in the training set by each model.

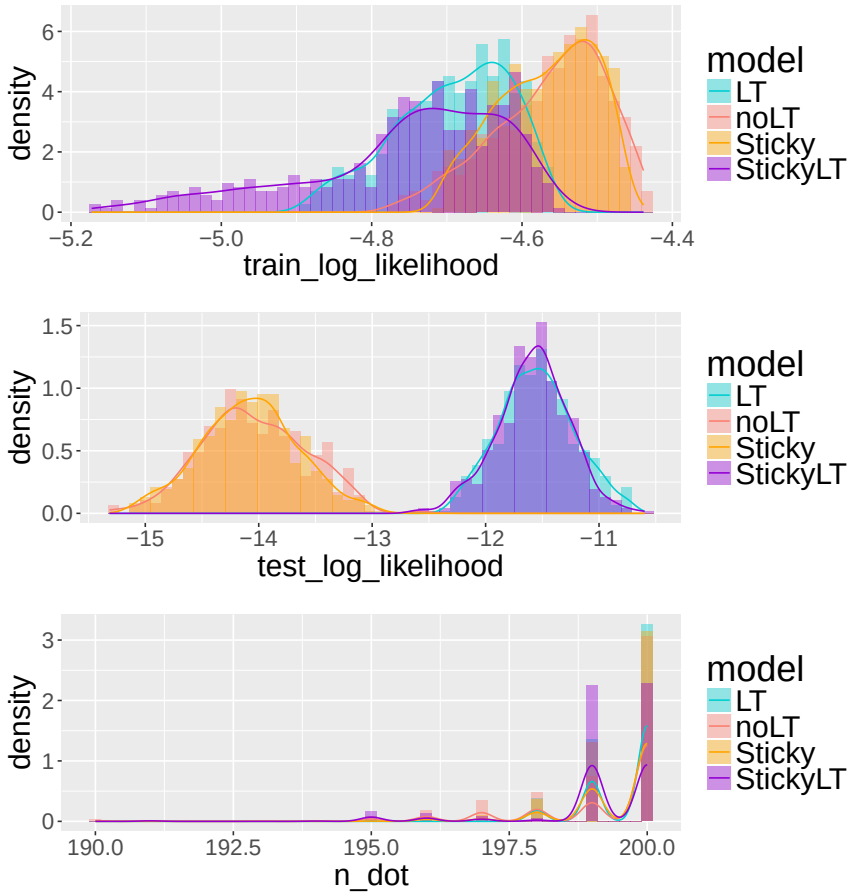


Figure A.26: Top and Middle: Training set and test set log marginal likelihoods for Bach chorale data on the four HDP-based models: HDP-HMM-LT, HDP-HMM, Sticky HMM, and Sticky HDP-HMM-LT, where the two LT models set $\lambda = 1.0$. Bottom: Number of latent states occupied in the training set by each model.

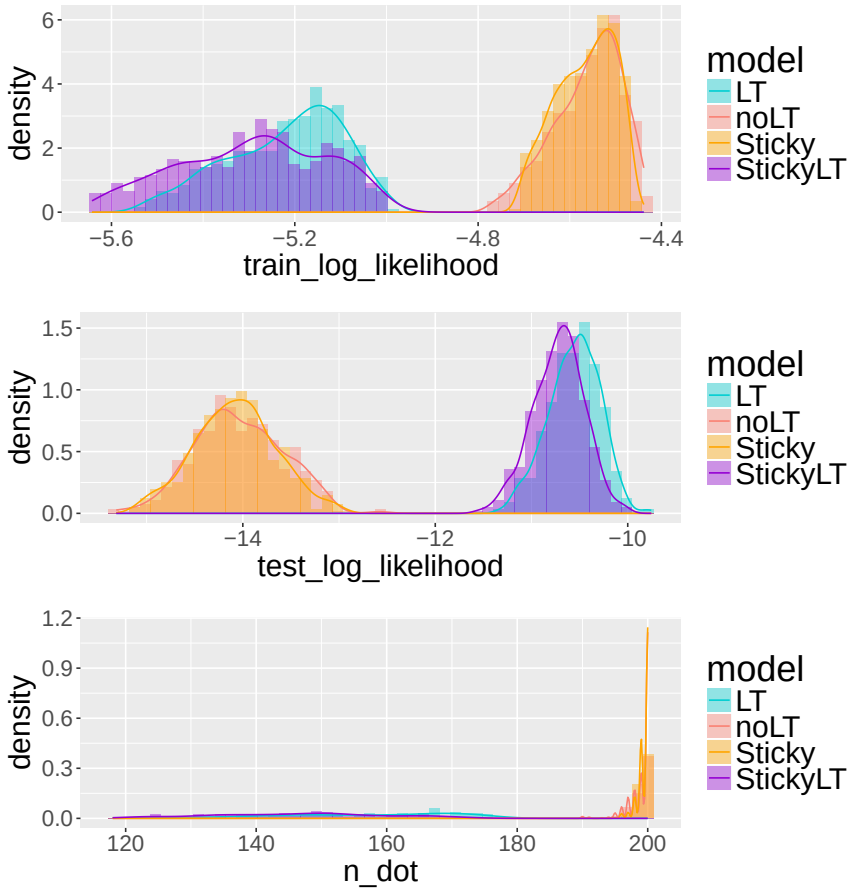


Figure A.27: Top and Middle: Training set and test set log marginal likelihoods for Bach chorale data on the four HDP-based models: HDP-HMM-LT, HDP-HMM, Sticky HMM, and Sticky HDP-HMM-LT, where the two LT models set $\lambda = 1.0$. Bottom: Number of latent states occupied in the training set by each model.

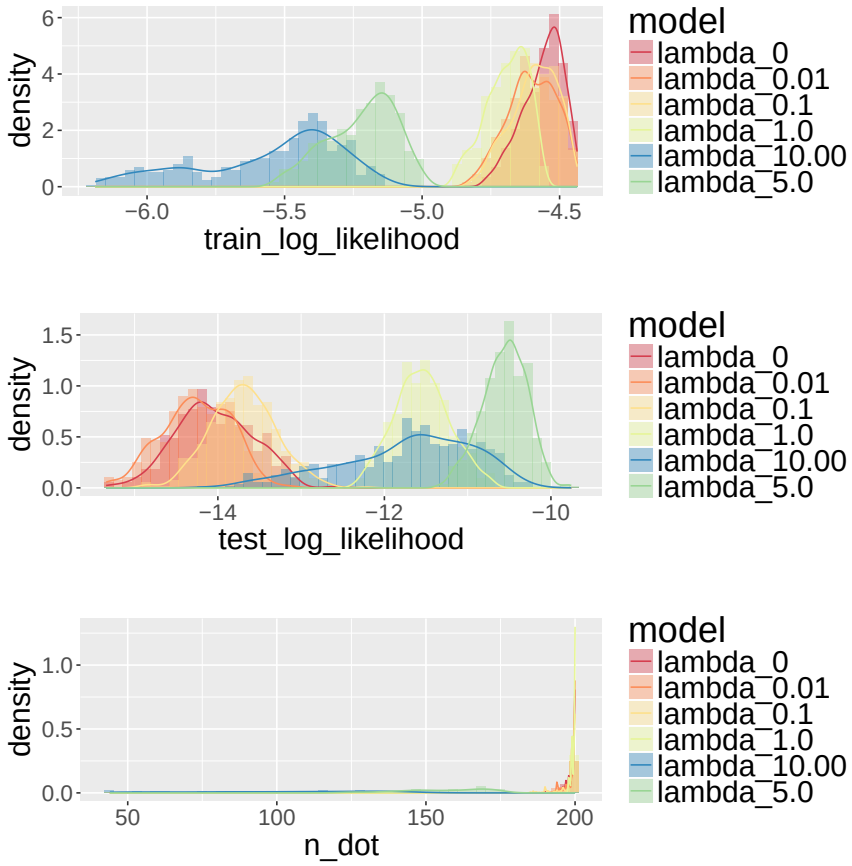


Figure A.28: Top and Middle: Training set and test set log marginal likelihoods for Bach chorale data on the four HDP-based models: HDP-HMM-LT, HDP-HMM, Sticky HMM, and Sticky HDP-HMM-LT, where the two LT models set $\lambda = 1.0$. Bottom: Number of latent states occupied in the training set by each model.

References

Akaike, H. (2011). Akaike's information criterion. In Lovric, M., editor, *International Encyclopedia of Statistical Science*, pages 25–25. Springer, Berlin.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. University of London London.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Box, G. E., Draper, N. R., et al. (1987). *Empirical model-building and response surfaces*, volume 424. Wiley New York.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158.

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

Fox, C. W. and Roberts, S. J. (2012). A tutorial on variational Bayesian inference. *Artificial intelligence review*, 38(2):85–95.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2008). An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM.

Gael, J. V., Teh, Y. W., and Ghahramani, Z. (2009). The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, pages 1697–1704.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 6:721–741.

Ghahramani, Z. and Griffiths, T. L. (2005). Infinite latent feature models and the Indian buffet process. In *Advances in neural information processing systems*, pages 475–482.

Ghahramani, Z., Jordan, M. I., and Smyth, P. (1997). Factorial hidden Markov models. *Machine learning*, 29(2-3):245–273.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pages 337–348.

Ishwaran, H. and Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87(2):371–390.

Johnson, M. and Willsky, A. S. (2014). Stochastic variational inference for Bayesian time series models. In *ICML*, pages 1854–1862.

Johnson, M. J. and Willsky, A. S. (2013). Bayesian nonparametric hidden semi-markov models. *The Journal of Machine Learning Research*, 14(1):673–701.

Kolter, J. Z. and Johnson, M. J. (2011). REDD: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, volume 25, pages 59–62. Citeseer.

Kurihara, K., Welling, M., and Teh, Y. W. (2007). Collapsed variational Dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801.

Lindsay, B. G. (1995). Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR.

McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

Neal, R. M. (2003). Slice sampling. *Annals of statistics*, pages 705–741.

Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Quick, D. (2014). Kulitta: a framework for automated music composition. *Unpublished doctoral dissertation, Yale University*.

Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *ieee assp magazine*, 3(1):4–16.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rasmussen, C. E. (2000). The infinite gaussian mixture model. In Solla, S., Leen, T., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems*, volume 12, pages 554–560. MIT Press.

Rosenblatt, M. et al. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464.

Sethuraman, J. (1994). A constructive definition of Dirichlet processes. *Statistica Sinica*, 4:639–650.

Teh, Y. W. (2011). Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476).

Tripuraneni, N., Gu, S., Ge, H., and Ghahramani, Z. (2015). Particle Gibbs for infinite hidden Markov models. In *Advances in Neural Information Processing Systems*, pages 2395–2403.

Valera, I., Ruiz, F., Svensson, L., and Perez-Cruz, F. (2015). Infinite factorial dynamical model. In *Advances in Neural Information Processing Systems*, pages 1657–1665.

Van Gael, J., Saatci, Y., Teh, Y. W., and Ghahramani, Z. (2008). Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1088–1095. ACM.

Wasserman, L. (2007). *All of nonparametric statistics*. Springer.