Supplementary Material: MCMC Inference Algorithm for the BP-HMM

Anonymous CVPR submission

Paper ID 6

Abstract

This document presents supplementary algorithmic details developed for the POCV 2012 submission "Nonparametric Discovery of Activity Patterns from Video Collections". We focus on MCMC inference algorithms for the Beta Process Hidden Markov Model (BP-HMM), incorporating novel data-driven proposals for birth/death moves in the reversible jump framework. Correct updates for sequence-specific and global dynamics parameters are also discussed.

1. Inference Overview

To perform inference for the BP-HMM, we develop a collapsed Markov Chain Monte Carlo (MCMC) algorithm based on that suggested by Fox et al [1]. The variables of interest instantiated in the Markov chain are the binary feature assignments F, the transition weights η, and the emission parameters θ. We marginalize over the time series state assignments z in the HMM to achieve faster mixing, as done in [1]. We also marginalize over feature inclusion parameters b_k implicit in the Beta Process, this allows using the collapsed Indian Buffet Process formulas for $p(F_{i,k}|F_{-i,k})$, which also improves mixing.

Here we provide details about the components of this MCMC algorithm that update the feature assignments F for all training videos given fixed values of other variables of interest. We also provide brief details about updating the transition weights η for both individual and global dynamics models. Given auxiliary variable state sequences $\mathbf{z}$ for all videos, updates for θ are simple, conjugate, closed-form draws from a Dirichlet posterior, since the likelihood $p(x|\theta)$ is multinomial and the prior is Dirichlet. As stated in the original paper, we assume constant values for all HBP hyperparameters: the mass parameter γ and concentration β.

2. Sampling Feature Assignments F

The updates to binary feature assignments F procede iteratively, altering one time-series at a time. For each time series, we divide its possible included features into two disjoint groups: those already possessed by some other time series (what we call *shared* features), and those unique to the current time series and possessed by no others (*unique* features). Resampling the features for the current time series procedes in two steps: updates for the shared features, and updates for unique features. These are discussed in sections 2.1 and 2.2 respectively.

2.1. Sampling Shared Features for Individual Time Series

The algorithm for sampling *shared* feature assignments for an individual time series is given in Alg. **??**. This procedure sweeps through all shared features and proposes flipping time series i's current binary indicator for that feature to its opposite value. Note that this implies we can only change one entry in the binary assignment vector f_i at a time.

The acceptance ratio for this move requires three terms in both numerator and denominator, as shown in Alg. **??** eq. 6. These terms are the prior probability of the features given hyperparameters, the likelihood of the HMM emissions x_i

given the features, and the probability of proposing the given move. We address each below.

We compute the prior term $p(f_{i,k} = 1|f_{-i,k}, \beta_k)$ in a straightforward way using the Indian Buffet Process. This yields the following expression, which makes use of sufficient statistics $m_k^{-i} = \sum_{i' \neq i} f_k^{i'}$ and $n_k^{-i} = \sum_{i' \neq i} 1$.

$$p(f_{i,j,k} = 1|f_{-i,k}, \beta, \gamma) \propto \frac{m_k^{-i}}{n_k^{-i} + \beta} \tag{1}$$

For the likelihood term, we must efficiently compute the marginal probability $p(x_i|f, \eta, \theta)$. This is accomplished via HMM dynamic programming which marginalizes over the possible state sequence assignments z_i for the current time series. We use efficient Matlab routines provided by the authors of [1] for this task, which compute $\log p(x_i|f, \eta, \theta)$. We denote the operator that can compute this log-likelihood as $\mathcal{M}(x_i, f, \eta, \theta)$.

In general, the acceptance ratio of any Metropolis-Hastings update also includes terms for the probability of proposing the considered move. Here, the proposal is symmetric, since we always flip a 0 to a 1 or vice versa. This means that the forward and reverse move terms cancel out, as illustrated in Alg. 1 equation 6.

2.2. Sampling Unique Features for Individual Time Series

The algorithm for sampling *unique* feature assignments for an individual time series is given in Alg. 2. We follow the reversible jump MCMC procedure in Fox et al. to accomplish sampling unique features. This procedure proposes either creating a single new feature for the current time series (a "birth" move), or deleting one of the existing features unique to the time series (a "death" move). We then compute an acceptance ratio for this proposal, and decide with probability proportional to this ratio whether or not to accept the proposal. For an introduction to reversible jump MCMC, see [2].

Here, we assume as in [1] that the features are indexed such that all K existing features possessed by at least one object are numbered $1, 2, \ldots K$. We further assume that a newly created feature is placed at index $K+1$. Thus, if k represents the location of some feature possessed only by object i, we denote a death move by $f_i - \delta_k$, wheren δ_k denotes the indicator vector that is all zeroes except at index k.

Below we give further details about the data-driven proposals we develop to improve acceptance of birth proposals, which differs significantly from the procedure in Fox et al.

2.2.1 Birth-Death Moves with Data-Driven Proposals

We must set a fixed probability for selecting between birth and death moves. We set $p_{birth} = 0.5$, and partition the remaining mass evenly over death moves at each of the existing unique features for the time series. In the edge case where no unique features exist in a time series, we always propose a birth move ($p_{birth} = 1$).

We improve upon the birth proposals suggested by Fox et al. by making the emissions parameters for a new feature data-driven rather than drawn from the prior. Specifically, our proposals select a subwindow of the current time series at random and propose a new feature whose emission distribution is a mixture of the posterior for all codewords in the window and the prior.

We must select a subwindow of the current time series i, which consists of timesteps $1, 2, \ldots T_i$. We define a subwindow by two parameters, its length W (the number of timesteps it contains), and its starting location w_0 in the current time-series. We randomly select a window length W and a starting position w_0 as follows

$$W \sim Unif(L_{min}, L_{max}) \tag{7}$$
$$w_0 \sim Unif(1, 2, \ldots T_i - W) \tag{8}$$

Algorithm 1 `SampleSharedFeaturesBP` : MCMC updates for shared features for time series i

Input:

i : object ID
F : current feature assignments for all objects
n_k^{-i} : count of other time series instances
$m_k^{-i} = \sum_{i' \neq i} F_{i',k}$: count of other instances possessing feature k
η : state-to-state transition weights for current object
θ : emission distribution parameters for each hidden feature k
$\mathcal{M}(f, \eta, \theta)$: function to compute marg. log prob. $\log p(x_{1:T_i}^{(i)} | f^{(i)}, \eta, \theta)$
β : concentration parameter of the BP

Output:

f_i : binary feature assignment vector for object i with shared features updated

Procedure:

1: Set current binary feature assignments f for object i

$$f = F_{i,:} \tag{2}$$

2: Build set of shared features active in objects other than i

$$\mathcal{S} = \{k : F_{i',k} = 1\} \text{ for } i' \neq i,\ k \in 1 \dots K \tag{3}$$

3: **for all** shared features $k \in \mathcal{S}$ **do**

4: Propose flipping binary feature assignment k from current value f to new value f^*

$$f^* = f \tag{4}$$
$$f_k^* = \neg f_k \text{ with probability } 1 \tag{5}$$

5: Compute acceptance ratio ρ for this proposal

$$\begin{aligned} \rho &= \frac{p(f_k^* | F_k^{-i}) \cdot p(x_{1:T}^{(i)} | f^*, \eta, \theta) \cdot \cancelto{1}{g(f_k | f_k^*)}}{p(f_k | F_k^{-i}) \cdot p(x_{1:T}^{(i)} | f, \eta, \theta) \cdot \cancelto{1}{g(f_k^* | f_k)}} \\ &= \left[\frac{m_{j,k}^{-i}}{n_{j,k}^{-i} - m_{j,k}^{-i} + \beta} \right]^{f_k^* - f_k} \cdot \exp\left[\mathcal{M}(x_i, f^*, \eta, \theta) - \mathcal{M}(x_i, f, \eta, \theta) \right] \end{aligned} \tag{6}$$

6: Accept or Reject the proposed feature assignment

$$F_{i,j,k} = \begin{cases} f_k^* & \text{with probability } \min(\rho, 1) \\ f_k & \text{o.w.} \end{cases}$$

7: **end for**

We allow the window lengths to vary between L_{min} and L_{max}, which are set to 1 and $\max(T_i, 50)$ respectively. We then build a histogram $h_{i,w_0,W}$ of codeword counts observed in the current window of the time series by sweeping over all timesteps in the window defined by W, w_0 and tallying up each codeword u seen at each timestep:

$$\mathbf{h}_i = [h_{i,1} \dots h_{i,V}]^T, \quad h_{i,v} = \sum_{t: w_0 \leq t \leq w_0 + W} \sum_{u \in \mathbf{x}_t} \delta_{u,v} \tag{9}$$

This creates a histogram $\mathbf{h}_i$ with V bins, one for each codeword in the codebook (we fix $V = 1000$ in all experiments). We then draw the proposal for emission parameters for new feature k' as a convex combination of the prior and the *posterior* over the chosen subwindow of the data.

$$\theta_{k'}|w_0, W \sim q_\theta(\cdot) = \frac{1}{2}\text{Dir}(\lambda_\theta \mathbf{u} + \mathbf{h}_i) + \frac{1}{2}\text{Dir}(\lambda_\theta \mathbf{u}) \tag{10}$$

We define this proposal as a mixture of prior and posterior (rather than just the posterior) in order to balance the need to accept new birth proposals with the need to accept death proposals when appropriate. In general, using any Metropolis-Hastings technique accepting a proposal requires that the "reverse" move is not too unlikely, since the accept ratio includes the factor $q(f \leftarrow f^*)/q(f^* \leftarrow f)$, and if $q(f \leftarrow f^*)$ (the reverse move term) is too small the accept rate will fall to zero. If we just used the posterior over the subwindow, we would rarely delete any behavior that didn't match the randomly chosen subwindow of the data. Using the prior as well as the posterior in the mixture proposal makes accepting both birth and death proposals reasonably likely.

Note that when drawing samples from $q_\theta(\cdot)$, we choose either posterior or prior with probability $\frac{1}{2}$, and then draw from the corresponding Dirichlet distribution using standard techniques. When evaluating $q_\theta(\cdot)$ in the acceptance ratio, however, we evaluate the probability under the full mixture.

Examining the above proposal, we note that this windowed proposal can be written as the result of a diffeomorphism (one-to-one, differentiable function) that is constant with respect to the current sampler state (F, η, θ). This means the Jacobian term required in the RJMCMC acceptance ratio [2] remains unity, just as with the proposal from the prior.

The overall probability of proposing a particular configuration of f^*, η^*, and θ^* after a birth move is given by

$$q_{birth}(f^*, \eta^*, \theta^* | f_i, \eta, \theta) = p_{birth} q_\theta(\theta^*) q_\eta(\eta^*) \tag{11}$$

where $q_\theta()$ is defined in equation 10 and q_η is simply the prior over η. Thus, in the final acceptance ratio the q_η terms always cancel out, but we must include the other terms in equation 11.

Note that the random choice of the subwindow for the data-driven proposal is made separately and independently from the choice of move type (birth or death). Because this choice of subwindow does not depend on the current move or the current state, its probability does not factor in to the acceptance ratio calculation. This choice is effectively a choice over random transition kernels, and a random selection among transitions is a valid way to construct an MCMC method, so long as the transitions are valid on their own [3].

3. Sampling Transition Weights η

3.1. Sequence-specific parameters $\eta^{(i)}$

Let $\eta_j^{(i)}$ denote the outgoing transition probability vector from state/behavior j.

Fox *et al*. provide the following posterior for $\eta_{j,:}^{(i)}$, which we find to be incorrect

$$\eta_j^{(i)} | \mathbf{z}_i \sim \text{Gamma}(N_k + \alpha + \delta_{j,k}\kappa, 1) \text{ (NOT CORRECT)} \tag{15}$$

The correct posterior for $\eta_j^{(i)}$ is given up to a proportionality constant as

$$p(\eta_{j,k}^{(i)} | \mathbf{z}_i, f_{i,k} = 1) \propto \frac{(\eta_{j,k}^{(i)})^{N_{j,k}^{(i)} + \alpha + \delta_{j,k}\kappa - 1} e^{-\eta_{j,k}}}{\left[\sum_{k': f_{i,k'}=1} \eta_{j,k'}\right]^{N_j^{(i)}}} \tag{16}$$

where $N_{j,k}^{(i)}$ counts the number of transitions from state j to k in sequence z_i, and $N_j^{(i)} = \sum_{k: f_{i,k}=1} N_{j,k}^{(i)}$.

Draws from this posterior can be obtained by sampling auxiliary variables

Algorithm 2 `SampleUniqueFeaturesBP` : MCMC updates for unique features for time series i

Input:

i : time series instance ID
x_i : observed codeword emissions for the current time series j : category label for time series i
F : current feature assignments for all time series objects in the corpus
N : count of all time series instances
η : state-to-state transition weights for current object/category
θ : emission distribution parameters for each hidden feature k
$\mathcal{M}(f, \eta, \theta)$: function to compute marg. log prob. $\log p(x^{(i)}_{1:T_i} | f^{(i)}, \eta, \theta)$
$q_\theta(\theta)$: proposal distribution that creates or deletes feature emission parameters
$p_\theta(\theta)$: prior distribution on emission parameters
$q_\eta(\eta)$: proposal distribution that creates or deletes transition weights (assumed to be prior on η)
γ : mass parameter for the BP
β : concentration parameter for the BP

Output:

F_i : feature assignments for time series i with unique features updated

Procedure:

1: Build set of features used only in object i

$$\mathcal{U}_i = \{k : F_{i,k} = 1 \text{ and } F_{i',k} = 0\} \text{ for } i' \neq i, k \in 1, 2, \dots K \tag{12}$$

2: Count number of current unique features: $U_i = |\mathcal{U}_i|$
3: Compute poisson rate parameter ν where $U_i \sim \text{Poisson}(U_i|\nu)$

$$\nu = \gamma \frac{\beta}{N - 1 + \beta} \tag{13}$$

4: Propose new value f^*_k at feature k. Either birth ($k = K + 1$) or death move at existing feature k. Always do birth if $U_i = 0$.

$$f^* = \begin{cases} f + \delta_{K+1} & \text{with prob. } p_{birth} \\ f - \delta_k & \text{with prob. } \frac{1-p_{birth}}{U_i} \text{ for each } k \in \mathcal{U}_i \end{cases} \tag{14}$$

5: Create or delete appropriate entries for η^*, θ^* via draws from $q_\eta(), q_\theta()$
6: Compute acceptance ratio for this proposal

$$\begin{aligned} \rho &= \frac{p(\mathcal{U}^*_i | F_{-i,\mathcal{U}^*_i} = 0, \gamma, \beta) \cdot p(x^{(i)}_{1:T} | f^*, \eta, \theta)}{p(\mathcal{U}_i | F_{-i,\mathcal{U}_i} = 0, \gamma, \beta) \cdot p(x^{(i)}_{1:T} | f, \eta, \theta)} \cdot \frac{p_\theta(\theta^*)}{p_\theta(\theta)} \cdot \frac{p_\eta(\eta^*)}{p_\eta(\eta)} \cdot \frac{q(f_k, \theta, \eta | f^*_k, \theta^*, \eta^*)}{q(f^*_k, \theta^*, \eta^* | f_k, \theta, \eta)} \\ &= \frac{\text{Poi}(U^*_i|\nu)}{\text{Poi}(U_i|\nu)} \exp\left[\mathcal{M}(x_i, f^*, \eta^*, \theta^*) - \mathcal{M}(x_i, f, \eta, \theta)\right] \frac{q(f_k|f^*_k)}{q(f^*_k|f_k)} \left[\frac{p_\theta(\theta^*_k)}{q_\theta(\theta^*_k)}\right]^{f^*_k} \left[\frac{q_\theta(\theta_k)}{p_\theta(\theta_k)}\right]^{f_k} \end{aligned}$$

7: Accept or Reject the proposed feature assignment

$$F_{i,k} = \begin{cases} f^*_k & \text{with prob. } \min(\rho, 1) \\ f_k & \text{o.w.} \end{cases}$$

$$\bar{\eta}^{(i)}_j \sim \text{Dir}(N^{(i)}_{j,k} + \alpha + \delta_{j,k}\kappa) \tag{17}$$

$$C^{(i)}_j \sim \text{Gamma}(K\alpha + \kappa, 1) \tag{18}$$

and then setting $\eta_{j,k}^{(i)} = C_j^{(i)} \bar{\eta}_{j,k}^{(i)}$. This procedure inverts the usual Gamma to Dirichlet scaling transformation.

3.2. Global dynamics parameters $\eta^{(0)}$

Given observed state sequences $\mathbf{z}_i$ for all video objects i, we have the following posterior

$$\eta_{j,k}^{(0)}|\mathbf{z} \propto \left[\prod_i \left[\frac{1}{\sum_{k:f_{i,k}=1} \eta_{j,k}}\right]^{N_j^{(i)}}\right] \prod_k \text{Gamma}(\eta_{j,k}|N_{j,k}^{(0)} + \alpha + \kappa\delta_{j,k}, 1) \tag{19}$$

where we define the sufficient statistics of $\mathbf{z}$ based on counts of observed transitions from j to k in sequence i:

$$\begin{aligned} N_{j,k}^{(i)} &= \sum_{t=1}^{T_i-1} \delta_{j,z_t^{(i)}} \delta_{k,z_{t+1}^{(i)}} \\ N_{j,\cdot}^{(i)} &= \sum_{t=1}^{T_i-1} \delta_{j,z_t^{(i)}} \\ N_{j,k}^{(0)} &= \sum_i N_{j,k}^{(i)} \end{aligned} \tag{20}$$

Due to the complicated coupling of the $\prod_i \frac{1}{\sum \eta^{(i)}}$ terms, this does not lend itself readily to closed-form updates. We employ a simple random-walk Metropolis Hastings scheme that updates each entry $\eta_{j,k}^{(0)}$ individually.

The proposal for a new value $\eta_{j,k}^*$ is a Gamma distribution with parameters a, b set such that the mean is the previous value and variance σ^2 is fixed to a constant (we find 10 works well in our experiments).

$$q(\eta_{j,k}^*|\eta_{j,k}) \sim \text{Gamma}(\eta_{j,k}^*|mean = \eta_{j,k}, var = 10) \tag{21}$$

The acceptance probability then becomes

$$\rho = \frac{p(\mathbf{z}|\eta^*)}{p(\mathbf{z}|\eta)} \cdot \frac{(\eta_{j,k})^{\xi^*-\xi-\alpha-\delta_{j,k}\kappa}}{(\eta_{j,k}^*)^{\xi-\xi^*-\alpha-\delta_{j,k}\kappa}} \cdot \frac{(\sigma^2)^{\xi}}{(\sigma^2)^{\xi^*}} \cdot \frac{\Gamma(\xi)}{\Gamma(\xi^*)} \cdot \exp\left[\eta_{j,k} - \eta_{j,k}^*\right] \tag{22}$$

where $\xi = \frac{\eta_{j,k}^2}{\sigma^2}$ and $\xi^* = \frac{(\eta_{j,k}^*)^2}{\sigma^2}$

References

[1] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Sharing features among dynamical systems with beta processes. In *NIPS*, 2010. 1, 2

[2] P. Green and D. Hastie. Model choice using reversible jump markov chain monte carlo. *http://www.stats.bris.ac.uk/ peter/papers/HastieGreenR1.pdf*, 2009. 2, 4

[3] L. Tierney. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–1762, 1994. 4