1 Appendix

Note: The LSGM parameter set, Φ, which appears in the Main Document is omitted here for conciseness.

1.1 Joint distribution

The joint distribution of an observation sequence $X_{1:T}$ and a state sequence $\Pi_{1:T}$ is given by the following expression. Equation 2 uses the LSGM independence assumptions.

$$\log P[X_{1:T}, \Pi_{1:T}] = \sum_{i=1}^{T} \Big(\log P[\pi_i | \Pi_{1:i-1}, X_{1:i-1}] + \log P[x_i | \pi_i, \Pi_{1:i-1}, X_{1:i-1}] \Big) = \tag{1}$$

$$\sum_{i=1}^{T} \Big(\log P[\pi_i | \pi_{i-1}, X_{1:i-1}] + \log P[x_i | \pi_i, X_{1:i-1}] \Big) = \tag{2}$$

$$\sum_{i=1}^{T} \Big(\log \gamma_{i-1}(\pi_{i-1}, \pi_i) + \log e_i(\pi_i) \Big) \tag{3}$$

1.2 Forward, backward, Viterbi recursionss

The forward recursion is derived as follows

$$e_{t+1}(\pi_{t+1}) \sum_{\pi_t} \alpha_t(\pi_t) \gamma_t(\pi_t, \pi_{t+1}) = \tag{4}$$

$$P[x_{t+1} | \pi_{t+1}, X_{1:t}] \sum_{\pi_t} P[X_{1:t}, \pi_t] P[\pi_{t+1} | X_{1:t}, \pi_t] = \tag{5}$$

$$P[x_{t+1} | \pi_{t+1}, X_{1:t}] \sum_{\pi_t} P[X_{1:t}, \pi_t, \pi_{t+1}] = \tag{6}$$

$$P[X_{1:t+1}, \pi_{t+1}] = \alpha_{t+1}(\pi_{t+1}) \tag{7}$$

The backward recursion is derived as follows (Equation 11 uses the LSGM independence assumptions)

$$\sum_{\pi_{t+1}} \beta_{t+1}(\pi_{t+1}) \gamma_t(\pi_t, \pi_{t+1}) e_{t+1}(\pi_{t+1}) = \tag{8}$$

$$\sum_{\pi_{t+1}} P[X_{t+2:T} | \pi_{t+1}, X_{1:t+1}] P[\pi_{t+1} | \pi_t, X_{1:t}] P[x_{t+1} | \pi_{t+1}, X_{1:t}] = \tag{9}$$

$$\sum_{\pi_{t+1}} P[X_{t+1:T} | \pi_{t+1}, X_{1:t}] P[\pi_{t+1} | \pi_t, X_{1:t}] = \tag{10}$$

$$\sum_{\pi_{t+1}} P[X_{t+1:T} | \pi_{t+1}, \pi_t, X_{1:t}] P[\pi_{t+1} | \pi_t, X_{1:t}] = \tag{11}$$

$$\sum_{\pi_{t+1}} P[X_{t+1:T}, \pi_{t+1} | \pi_t, X_{1:t}] = \tag{12}$$

$$P[X_{t+1:T} | \pi_t, X_{1:t}] = \beta_t(\pi_t) \tag{13}$$

The Viterbi recursion for the LSGM may be derived as follows. Equation 15 uses the LSGM independence assumptions.

$$P_{max}\big[X_{1:t+1}, \pi_{t+1}\big] = \tag{14}$$

$$\max_{\pi_t} P_{max}\big[X_{1:t}, \pi_t\big] P[\pi_{t+1} | \pi_t, X_{1:t}] P[x_{t+1} | \pi_t, X_{1:t}, \pi_{t+1}] =$$

$$\max_{\pi_t} P_{max}\big[X_{1:t}, \pi_t\big] P\big[\pi_{t+1} | \pi_t, X_{1:t}\big] P\big[x_{t+1} | \pi_{t+1}, X_{1:t}\big] \tag{15}$$

Defining the Viterbi recursion variable as $\tau_t(k) = P_{max}\big[X_{1:t}, \pi_t = k\big]$, and locating the transition and emission terms in Equation 15, we obtain

$$\tau_{t+1}(k) = \max_j \tau_t(j)\gamma_t(j,k)e_{t+1}(k) \tag{16}$$

1.3 Modified forward recursion for length > 1 emissions

For a state emitting l-symbols at once (such as the motif state), the following recursion replaces the standard forward recursion. Here we assume that the state π_i emits symbol x_i, as before, however the state may emit more than one symbol as a group - in which case x_i is the last symbol emitted as a group.

$$\begin{aligned}
&P\big[X_{1:i}, \pi_i = j\big] \\
&= \sum_{\pi_{i-l}} P\big[X_{1:i-l}, \pi_{i-l}\big] P\big[\pi_i = j | \pi_{i-l}, X_{1:i-l}\big] \\
&\qquad P\big[X_{i-l+1:i} | \pi_i = j, X_{1:i-l}, \pi_{i-l}\big] \\
&= \sum_{\pi_{i-l}} P\big[X_{1:i-l}, \pi_{i-l}\big] P\big[\pi_i | \pi_{i-l}, X_{1:i-l}\big] \\
&\qquad P\big[X_{i-l+1:i} | \pi_i, X_{1:i-l}\big] \qquad (17) \\
&= \sum_{\pi_{i-l}} P\big[X_{1:i-l}, \pi_{i-l}\big] P\big[\pi_i | \pi_{i-l}, X_{1:i-l}\big] \\
&\qquad P\big[X_{i-l+1:i} | \pi_i\big] \qquad (18)
\end{aligned}$$

Equation 17 gives the modified forward recursion for the first-order full LSGM. Equation 18 provides the recursion for a half LSGM such as used in our TF-binding experiments.

1.4 Additional details

1.4.1 Multidimensional synthetic data experiments

The HMM has 3 states each emitting 4-dimensional full-covariance Gaussians. This requires $3 \times (4^2 + 4) = 60$ parameters. In addition, it uses 6 transition probabilities and 2 start probabilities bringing the total parameter count to 68.

The LSGM uses 65 parameters which are provisioned as follows.

- 1 GRU layer with 1 neuron without bias requiring $3 \times (4 \times 1 + 1^2) = 15$ parameters. A GRU layer without bias needs $3(mn + n^2)$ parameters for m inputs n outputs (as implemented in PyTorch version 0.4).
- 1 dense neuron taking the GRU output as input and providing a single output. Requires $1 \times 1 + 1 = 2$ parameters.
- Emission layer accepting the dense neuron's output and providing means and diagonal covariances for 3 states. The covariances use a single bias value per state. This requires $1 \times 12 + 12$ parameters for mean and $1 \times 12 + 3$ parameters for variance for a total of 39 parameters
- Transition layer accepting the dense neuron's output and providing 9 outputs representing the transition matrix. The transition network doesn't use biases. This needs $1 \times 9 = 9$ parameters

We performed the experiments over five times for each case, and took the average performance on the test set.

1.4.2 Multidimensional synthetic data generator parameters

- $\vec{\mu_1} = (0,0,0,0)^T, \vec{\mu_2} = (1,1,1,1)^T, \vec{\mu_3} = (-1,-1,-1,-1)^T$
- $\vec{\sigma_1^2} = \vec{\sigma_2^2} = \vec{\sigma_3^2} = (0.01, 0.01, 0.01, 0.01)$

- The diagonal covariance from f is limited to between $0.01, 0.04$ using sigmoid activation and scaling. f uses 33 GRU units
- $a_{pq} = \{0.7,\ if\ p = q,\ 0.15\ otherwise\}$

1.4.3 Synthetic scalar sequence generation

The algorithm for generating scalar synthetic sequences is given below. After this procedure is run, numpy (numpy.mean, numpy.var) is used to determine mean and standard deviation of all the datapoints in all the sequences, and then the sequences are normalized as $\frac{synthetic-\mu}{\sigma}$

Algorithm 1 Synthetic scalar data generation

procedure CHOOSETRANSITION($runningSum$)
  $\alpha = 10$
  $\beta = 30$
  if $runningSum < \alpha$ **then**
    $\vec{p} = [0.9, 0.05, 0.05]^T$
  end if
  if $\alpha \leq runningSum < \beta$ **then**
    $\vec{p} = [0.1, 0.8, 0.1]^T$
  end if
  if $\beta \leq runningSum$ **then**
    $\vec{p} = [0.05, 0.05, 0.9]^T$
  end if
  return $\vec{p}$
end procedure
procedure EMIT($seed$,$state$)
  $K = 1.2$
  if $state = 0$ **then**
    return $seed \times K$
  end if
  if $state = 1$ **then**
    return $seed$
  end if
  if $state = 2$ **then**
    return $seed/K$
  end if
end procedure
procedure GENSYNTHETICDATA($Length$)
  $synthetic$ = empty list
  $seed$ = random between $1e-2, 1$
  $counter = 0$
  for $0 < counter < Length$ **do**
    if $counter > 0$ **then**
      r = sum of last 10 items in $synthetic$
      $\vec{p}$ = CHOOSETRANSITION(r)
    else
      $\vec{p} = [1/3, 1/3, 1/3]^T$
    end if
    $state$ = Sample from multinomial distribution $\vec{p}$
    $seed$ = EMIT($seed$,$state$)
    Add $seed$ to $synthetic$
  end for
  return $synthetic$
end procedure

1.4.4 Details of experiments with synthetic scalar sequences

We trained HMMs with 3 to 60 states on 1000 sequences of length 25 generated using the procedure above using Expectation Maximization, similar to the method for Multidimensional synthetic data. The model that achieved the best BIC had 52 states. This corresponds to 52×51(transitions) $+52 \times 2$ (means, variances; emissions) + 51 (start probability) = 2807 parameters. We selected this model and reran the experiment with the model five times that gave an average log-likelihood of 1.416. This value was relatively stable across the five experiments.

We trained a single LSGM model with 2 GRU units and 10 states following the same architecture as for the case of multidimensional synthetic data. This model has a total of only 284 parameters, almost a tenth the size of the HMM. The model gave an average log-likelihood per data point of 2.259 with a sample standard deviation of 0.662 over five runs. While the model seems to be sensitive to initialization, it also provided better performance on average with a significantly lower parameter count.

1.4.5 Details of architecture search for TF-binding experiment

The following architectures were searched for the TF-binding experiment.

Table 1: Model parameters explored

Parameters	DeepBind	DeeperBind	LSGM	HMM
Num motifs	5,16	5	5	5
Motif length	11,24	11	11	11
LSTM layers	NA	1,2	2	NA
LSTM units	NA	20,32,64,128	64,128	NA
Dense layers	2	1,2	2	NA
Dense units	32	128,256	=LSTM units	NA
$\mathcal{F}(x)$	NA	NA	$exp(x), (ReLU(x))^2, (ReLU(x))^4$	=LSGM

1.4.6 Music modeling LSGM architecture

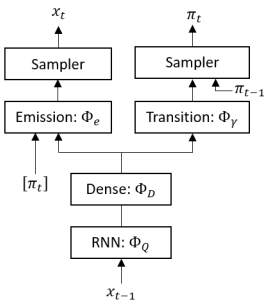

Figure: Music model architecture

Dataset / Network	Nottingham	Piano-midi	Muse	JSB
Recurrent	512	512	512	128
Dense[2]	512	512,512[1]	512,512	256,256
Emission	1536	1536	1536	1024
Transition	1024	1024	1024	512

[1]Comma-separated values represent multiple hidden layers, e.g., for Piano-midi, the model has two layers in it's Dense network
[2]Dense, Emission, Transition are fc networks
All cases except JSB use 16 LSGM states; JSB uses 32 LSGM states

Table: Layer sizes in music model