# Employing hidden Markov models of neural spike-trains toward the improved estimation of linear receptive fields and the decoding of multiple firing regimes

Sean Escola

Center for Theoretical Neuroscience

# LNP Model Overview

- Linear-nonlinear-Poisson
  - Stimuli are passed through a linear filter
    - The linear filter is the neuron's receptive field
  - The result is passed through a nonlinearity to determine the instantaneous firing rate
  - The rate defines an inhomogeneous Poisson process

$$\lambda(t) = f_\phi\left(\vec{k} \cdot \vec{s}(t)\right)$$

$$p(spike \text{ in } t, t+dt) = \lambda(t)\,dt$$

$$p(no\ spike) = e^{-\lambda(t)dt}$$

# LNP Model Learning

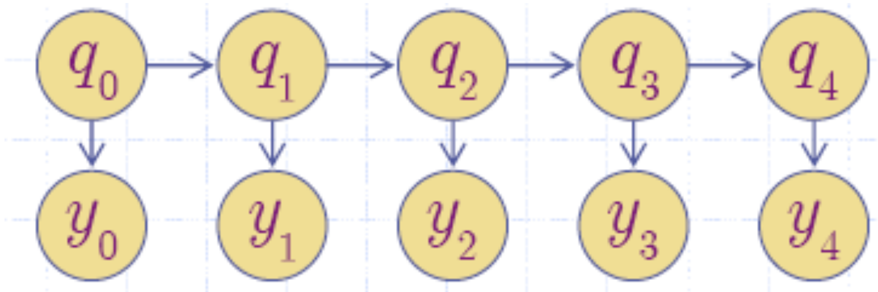
- Model parameters are learned by maximizing the log-likelihood
  - If $f_\phi$ is convex and log-concave, there is a unique solution easily found by gradient ascent
  - It doesn't matter your $f_\phi$ is wrong

$$L(\theta) \sim \sum_{i \in spikes} \log f_\phi\left(\vec{k} \cdot \vec{s}\,(t_i)\right) - \int f_\phi\left(\vec{k} \cdot \vec{s}\,(t)\right) dt$$

# HMM Overview

- Bayesian network model
- 2 events at every time-step (discrete model)
  - Transition to next step (hidden)
  - Emit observable (known)
- Emission probability distributions vary from state to state (in general)
  - Therefore, can inferred underlying sequence of hidden states from sequence of observables
- Uses the Markov assumption
  - The future is independent of the past given the present

- Graphical model:




- Markov assumption:

$$future \perp\!\!\!\perp past \mid present$$
$$p\left(q_t \mid q_{t-1}, q_{t-2}, \ldots, q_1, q_0\right) = p\left(q_t \mid q_{t-1}\right)$$

- Factorized complete probability distribution $X_U = \left\{y_{(0,T)}, q_{(0,T)}\right\}$:

$$p\left(X_U\right) = p\left(q_0\right)\prod_{t=1}^{T} p\left(q_t \mid q_{t-1}\right)\prod_{t=0}^{T} p\left(y_t \mid q_t\right)$$

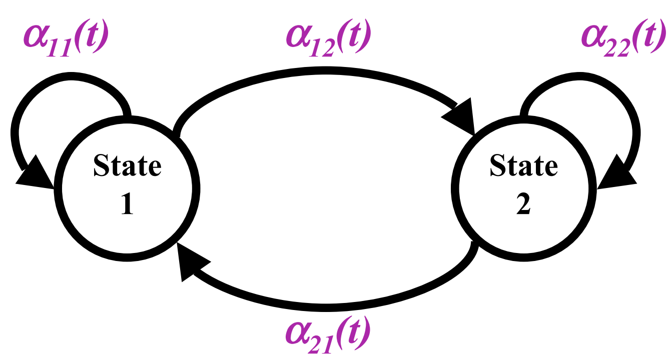
Images courtesy of T. Jebara, Dept. of CS, Columbia U.

# Improving RF estimation

- Original question:
  - Can we improve receptive field estimation by categorizing spikes as either informative or not informative about the stimulus, and then only use the relevant spikes to calculate the RF?
- HMM version:
  - Can we learn when the neuron is in a stimulus attentive state and when it is in some 'other' state?

# A 2-state model

- State 1: Attending to stimulus
- State 2: Attending to “other” activity
- Biologically reasonability:
  - UP/DOWN states
  - Tonic-burst LGN neurons



- Transition matrix:
  - Defined by rates is in LNP model
  - $\alpha(t) = \begin{pmatrix} e^{-g_{\varphi}\left(\vec{k}'_{12}\cdot\vec{s}(t)\right)dt} & g_{\varphi}\left(\vec{k}'_{12}\cdot\vec{s}(t)\right)dt \\ g_{\varphi}\left(\vec{k}'_{21}\cdot\vec{s}(t)\right)dt & e^{-g_{\varphi}\left(\vec{k}'_{21}\cdot\vec{s}(t)\right)dt} \end{pmatrix}$
- Emission matrix:
  - LNP model for state-1 (stimulus dependent)
  - Homogeneous Poisson process for state-2
  - $\eta(t) = \begin{pmatrix} f_{\phi}\left(\vec{k}_s\cdot\vec{s}(t)\right)dt & e^{-f_{\phi}\left(\vec{k}_s\cdot\vec{s}(t)\right)dt} \\ \lambda_o dt & e^{-\lambda_o dt} \end{pmatrix}$
- Model parameters:
  - $\vec{k}'_{12}$
    - Linear filter for transitioning while in state-1
  - $\vec{k}'_{21}$
    - Linear filter for transitioning while in state-2
  - $\vec{k}_s$
    - The neuron’s receptive field

# HMM Max Likelihood

- Since we don't know the hidden variables, we can't maximize the complete log-likelihood
- Incomplete likelihood:
  - $L(\theta) = \log \sum_{q_{(0,T)}} \pi_{q_0} \prod_{i=1}^{T} \alpha_{q_{i-1} q_i}(t_i) \prod_{i=0}^{T} \eta_{q_i y_i}(t_i)$
  - Exponential in *T*
- Use Expectation-Maximization
  - E-step: Guess the $q_i$'s at the current parameter settings (Baum-Welch)
  - M-step: Maximize the complete log-likelihood using the guessed $q_i$'s
  - EM is guaranteed to monotonically increased the incomplete log-likelihood
  - M-step is concave if $g_{\varphi}$ and $f_{\phi}$ are convex and log-concave

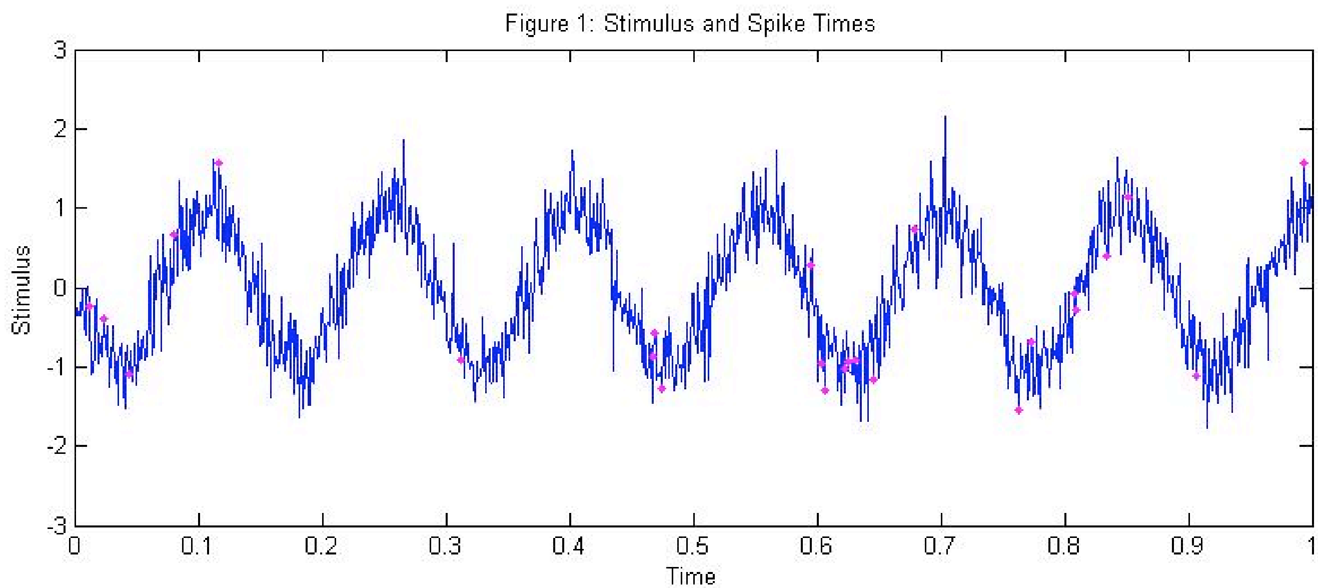
# Algorithmic considerations

- Convergence can be slow
  - EM can have linear convergence near the maximum likelihood solution
    - Usually quadratic convergence
  - It is possible to perform gradient ascent directly on the log-likelihood
    - The exact gradient can be calculated
    - This provides quadratic convergence
  - The best approach is to switch between EM and GA depending on the local likelihood landscape
  - Using a continuous time formulation can also help
    - In discrete time the probabilities for all time-steps must be calculated
      - You can easily have data with > 1e7 time-steps
      - Almost all time-steps have no associated spikes
    - In continuous time the probabilities are integrated from spike-time to spike-time
      - This involves much less computation and memory

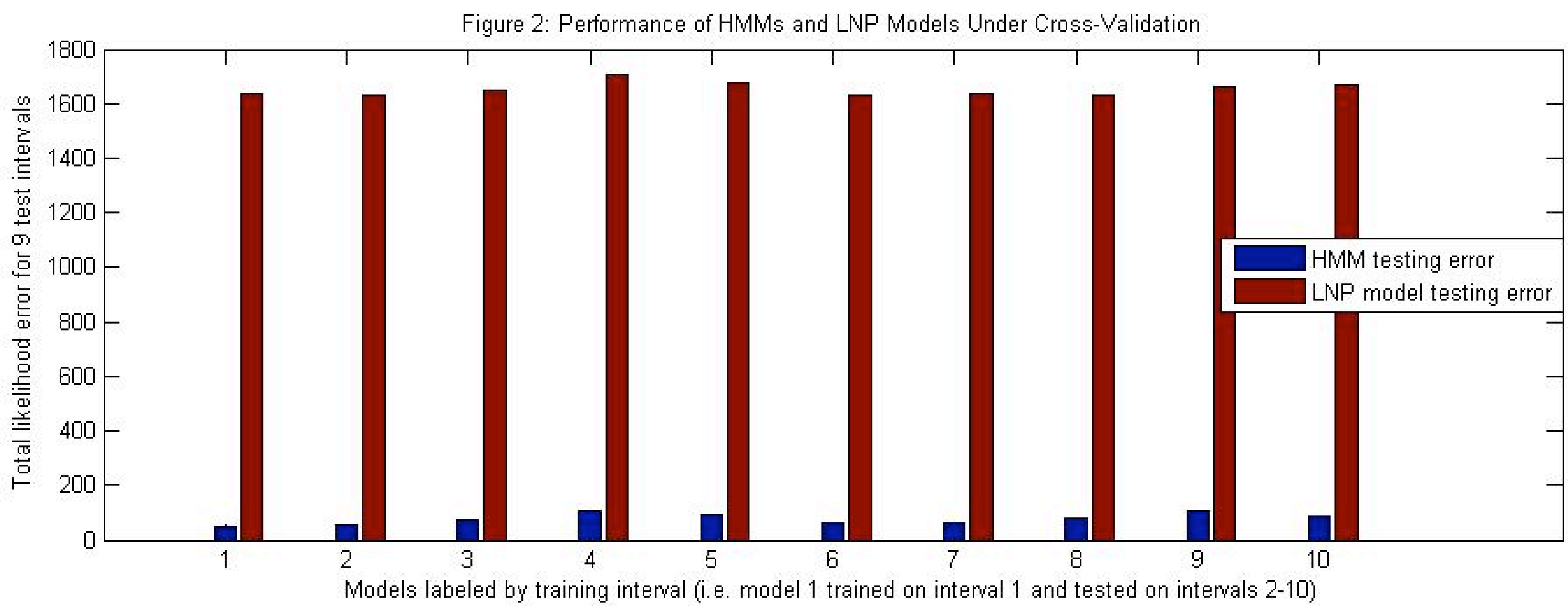
- Computations can be numerically unstable
  - Bernoulli approximation to Poisson distribution may fail when transition and firing rates get too high
    - $\lambda(t)dt + e^{-\lambda(t)dt} \neq 1$
  - It's okay to use to true Poisson distribution for firing (i.e. you can have more than 1 spike in a time-step)
    - $\eta(t_i) = \dfrac{\left(\lambda(t_i)dt\right)^{y_i} e^{-\lambda(t_i)dt}}{y_i!}$
  - The transition probabilities must be changed since it makes no sense to have more than 1 transition in a single time-step
    - $\alpha(t) = \begin{pmatrix} \dfrac{1}{1+g_\varphi\left(\vec{k}'_{12}\cdot\vec{s}(t)\right)dt} & \dfrac{g_\varphi\left(\vec{k}'_{12}\cdot\vec{s}(t)\right)dt}{1+g_\varphi\left(\vec{k}'_{12}\cdot\vec{s}(t)\right)dt} \\ \dfrac{g_\varphi\left(\vec{k}'_{21}\cdot\vec{s}(t)\right)dt}{1+g_\varphi\left(\vec{k}'_{21}\cdot\vec{s}(t)\right)dt} & \dfrac{1}{1+g_\varphi\left(\vec{k}'_{21}\cdot\vec{s}(t)\right)dt} \end{pmatrix}$
    - To guarantee concavity of M-step, $g_\varphi$ must grow exponentially
  - A continuous time formulation also solves this problem since it guarantees that the Bernoulli approximation is correct
  - As dt->0, the new discrete formulation and the continuous formulation are equivalent

# Preliminary Results

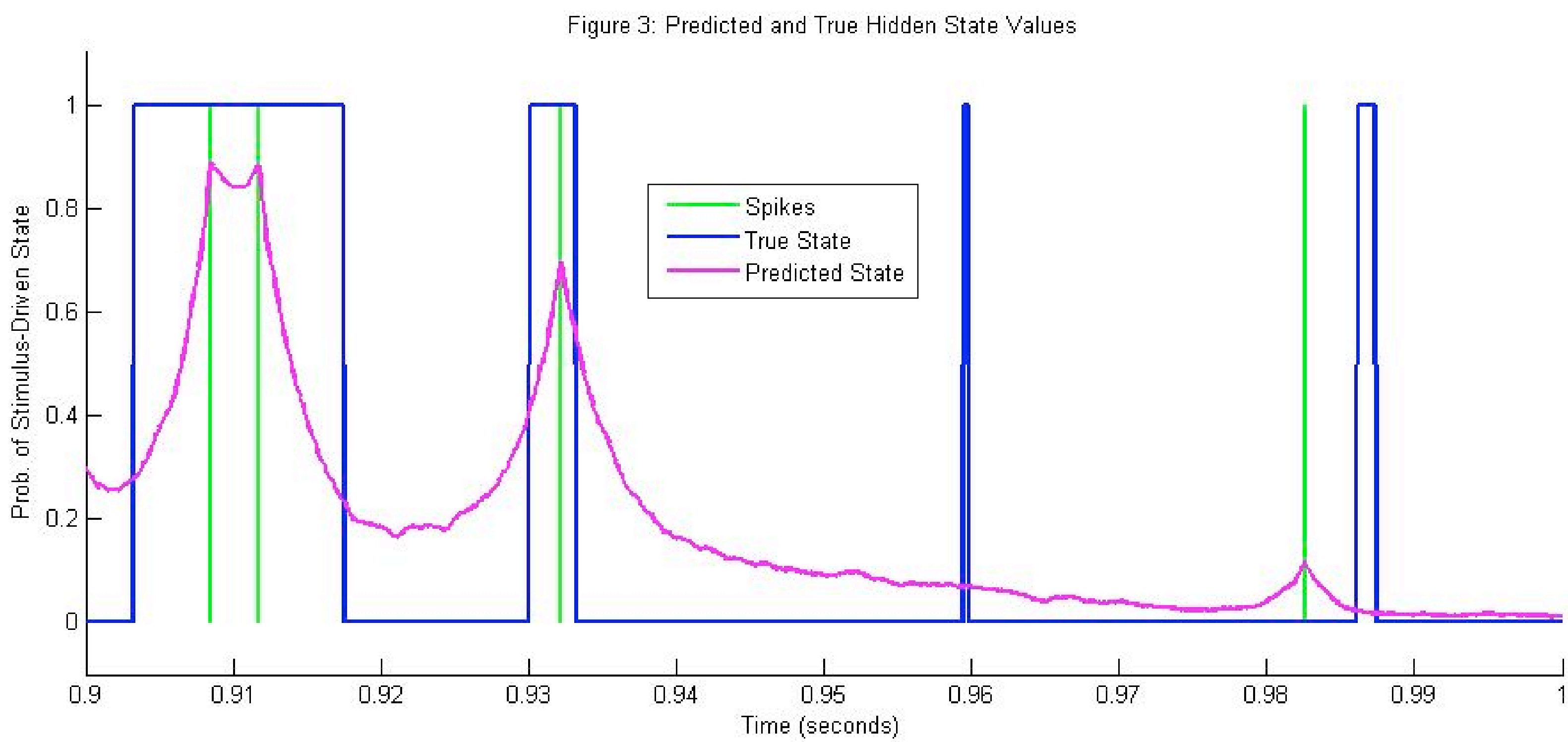
- I simulated 100 seconds of data
  - 1d, noisy sine-wave stimulus
  - $\vec{k}'_{12}$ likes positive stimulus values
  - $\vec{k}'_{21}$ , $\vec{k}_s$ like negative stimulus values
  - Firing rate in state-1: ~20 Hz
  - Firing rate in state-2: 10 Hz
  - All nonlinearity were the exponential $e^u$

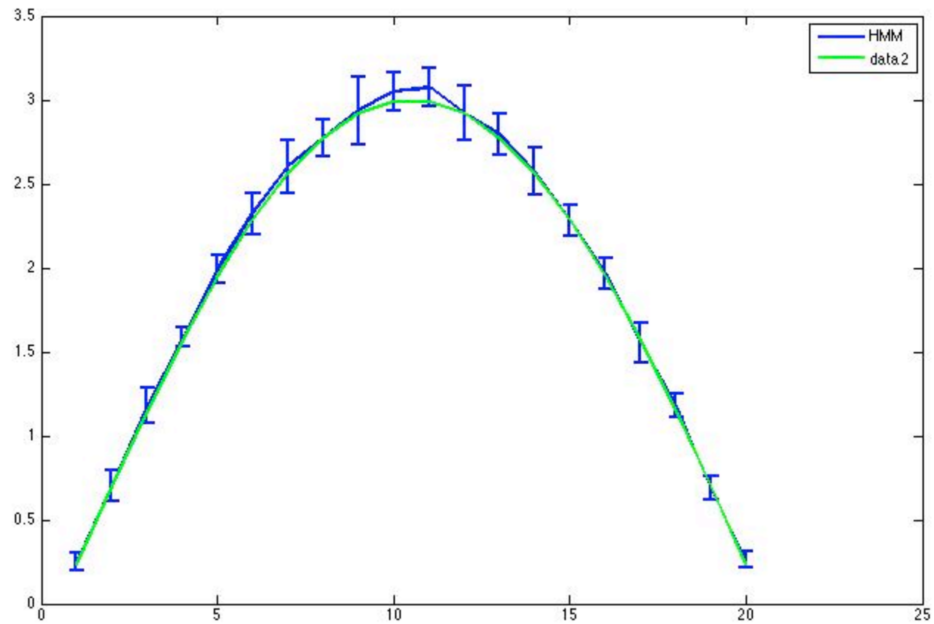


- The data were partitioned into ten 10 second segments.
- 10 HMMs and 10 standard LNP models were trained, 1 on each segment
- The remaining 9 segments were used to test the models
- The log-likelihoods shown are the total difference while testing from the log-likelihood achieved by the HMM trained on that segment
- All the HMMs outperformed all the LNP models on all segments

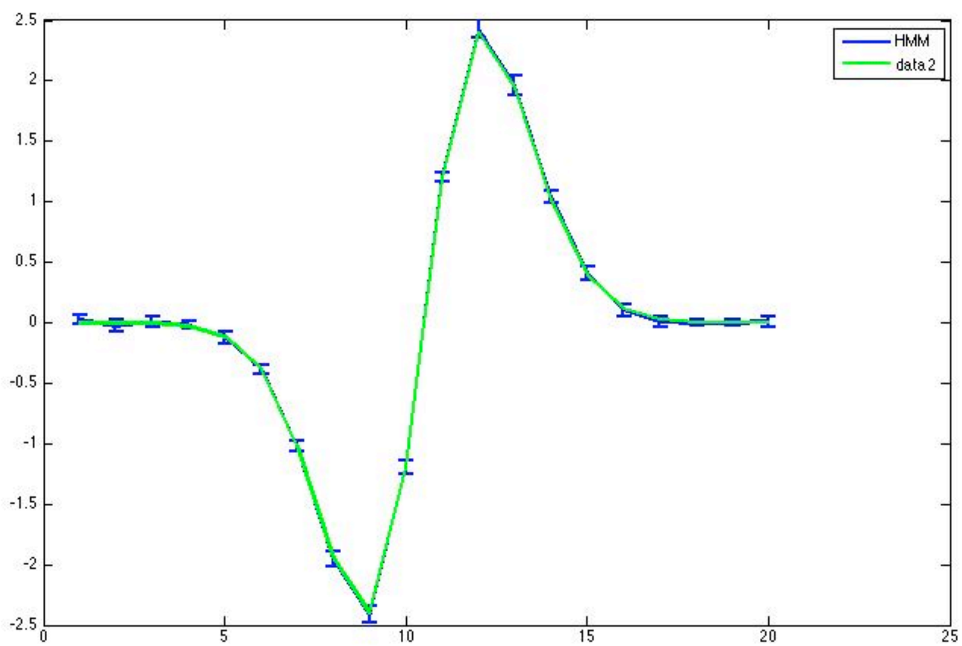


- After training, the inferred hidden state values show that the model did learn to distinguish state-1 from state-2
- There are 4 important combinations to predict
- Trivially (since the average firing rate is higher in state-1, and state-2 is more common):
  - Being in state-1 and spiking (i.e. 0.91 s)
  - Being in state-2 and not-spiking (i.e. 0.97 s)
- Not-trivially:
  - Being in state-2 and spiking (i.e. 0.98 s)
  - Being in state-1 and not spiking (shown elsewhere)

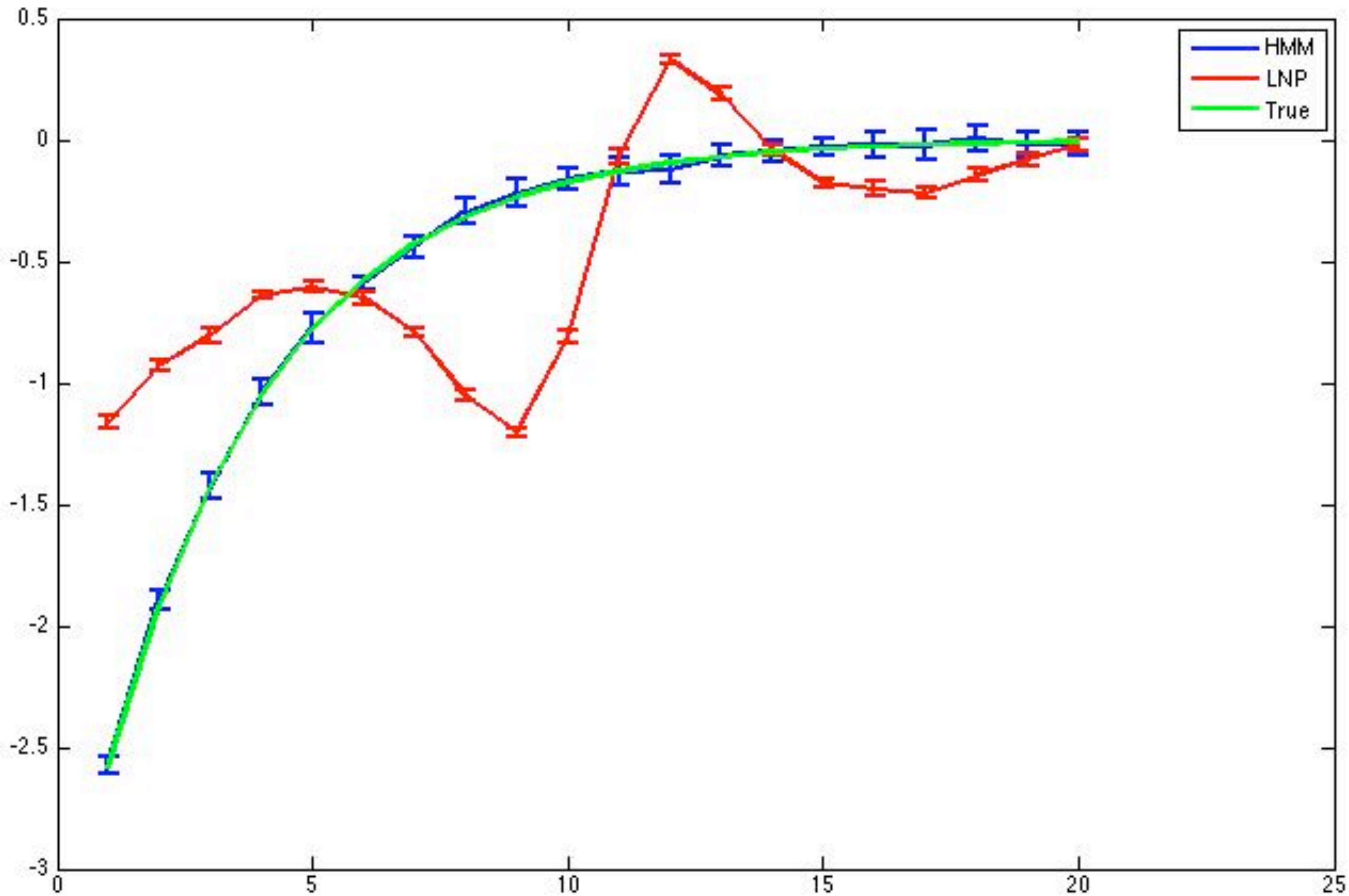




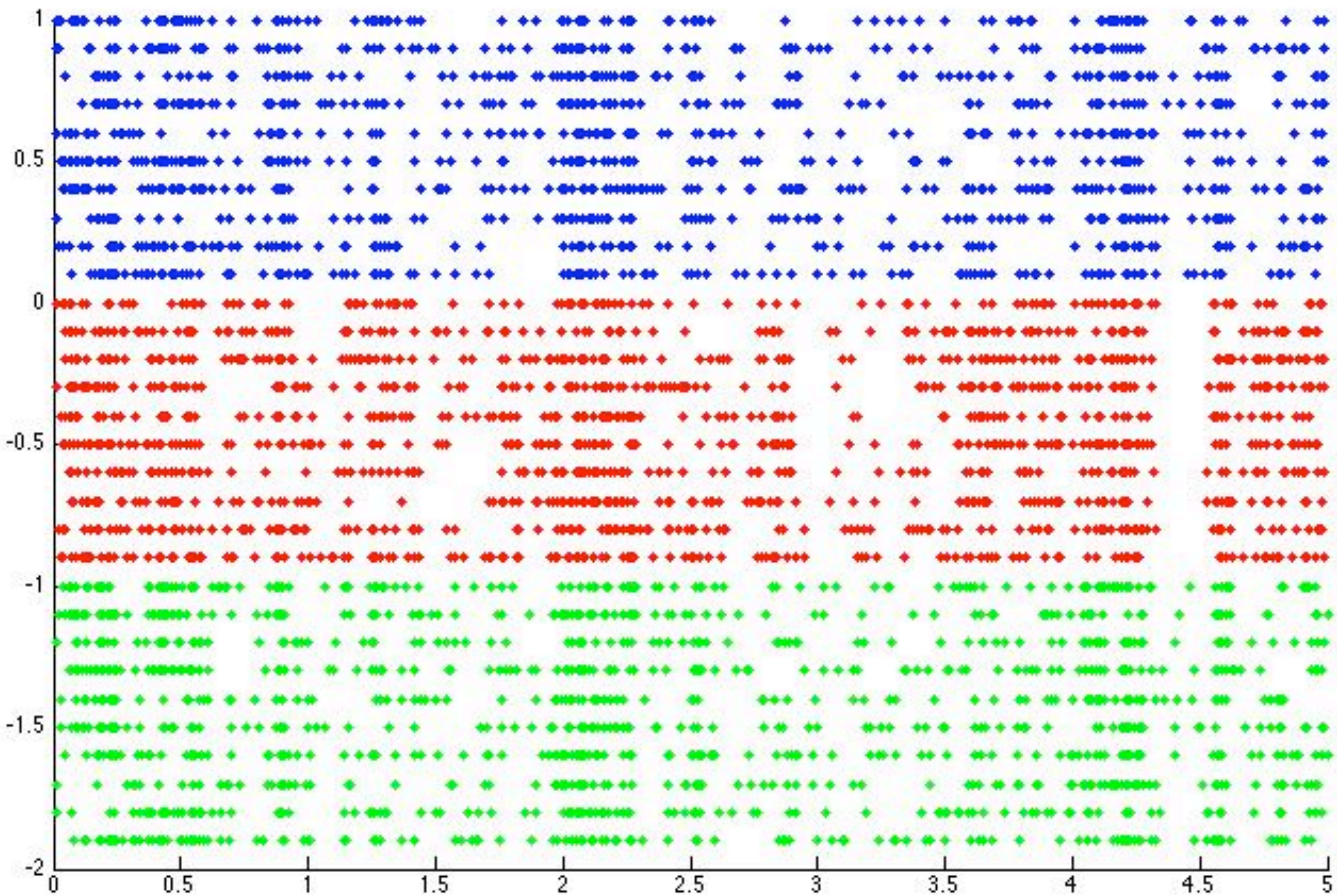
HMM
data2

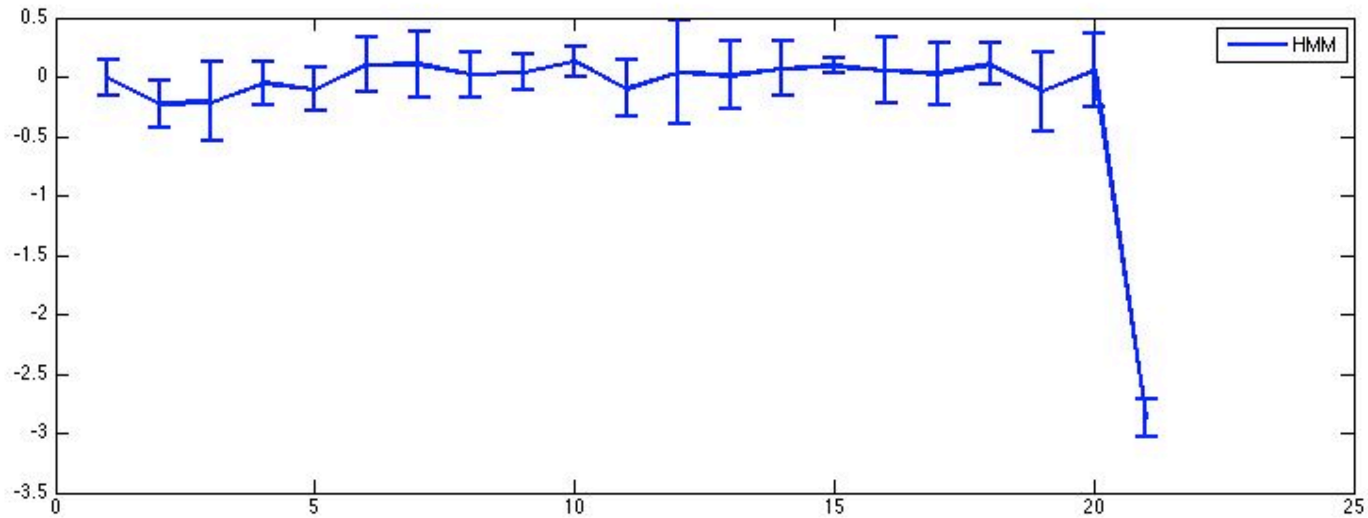



HMM
data2

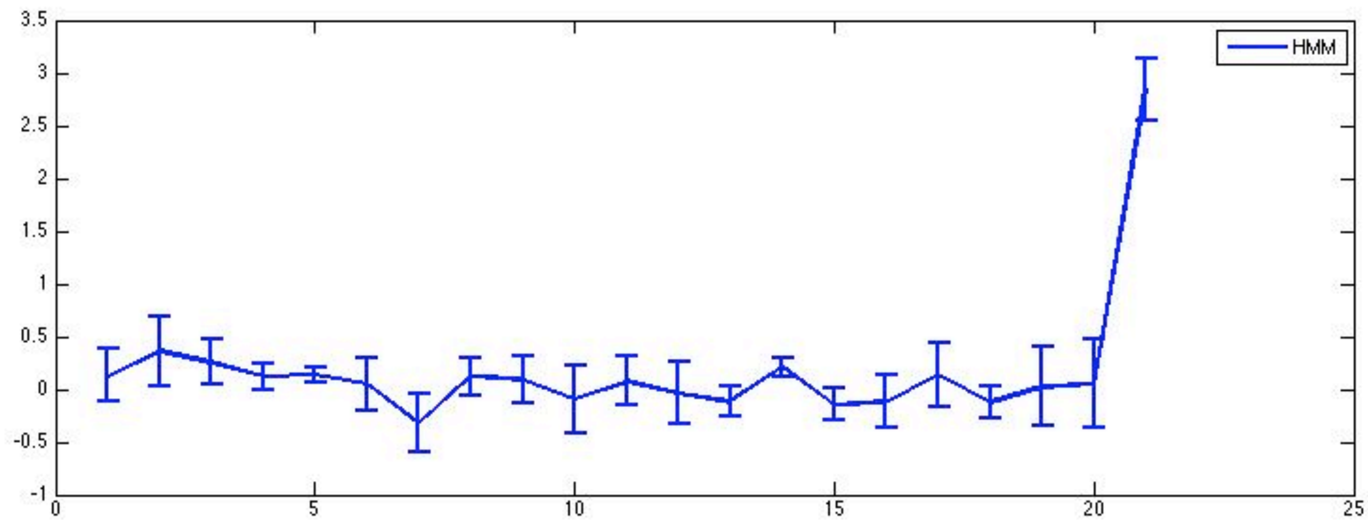


HMM
LNP
True
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
5
10
15
20
25

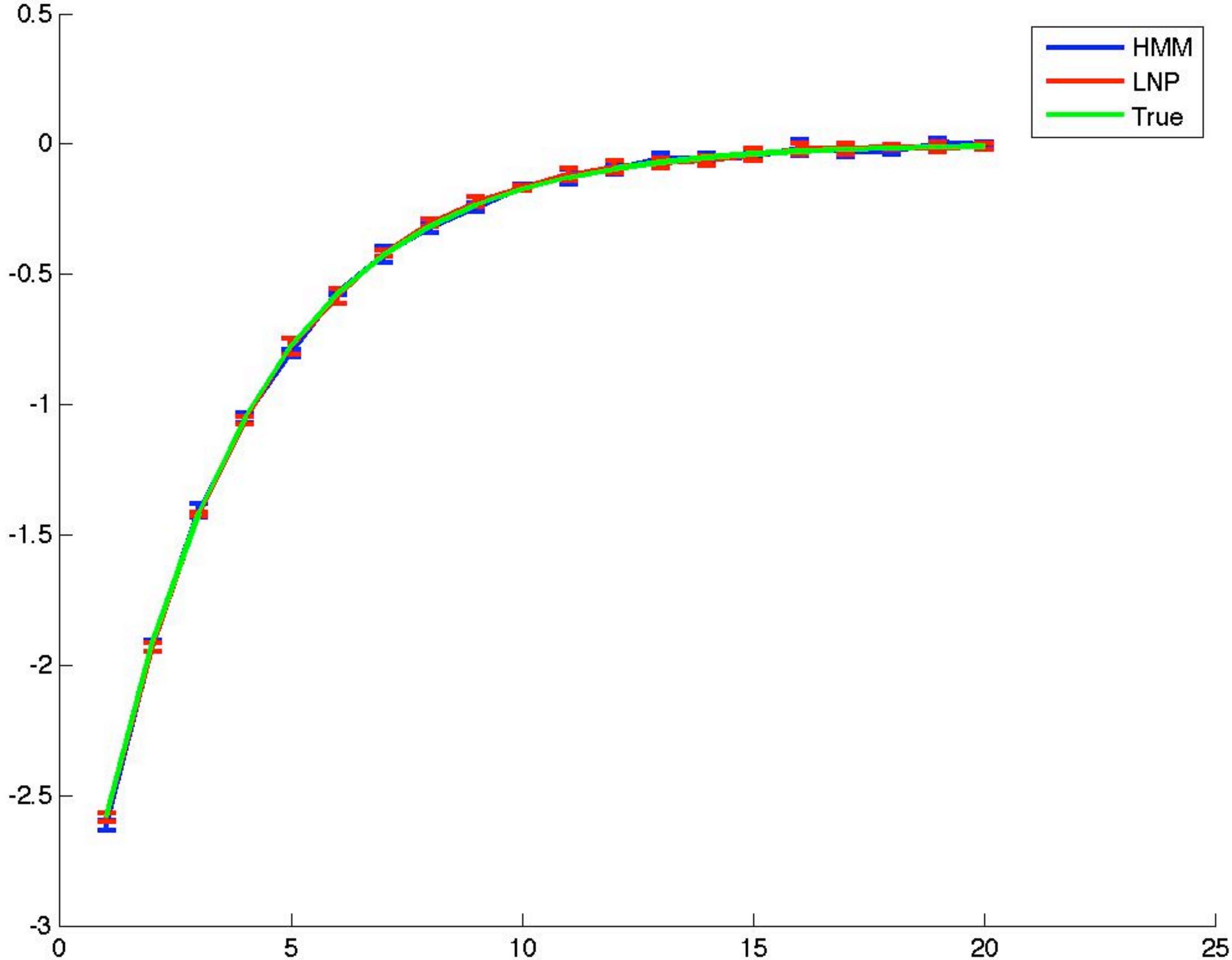

1
0.5
0
-0.5
-1
-1.5
-2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



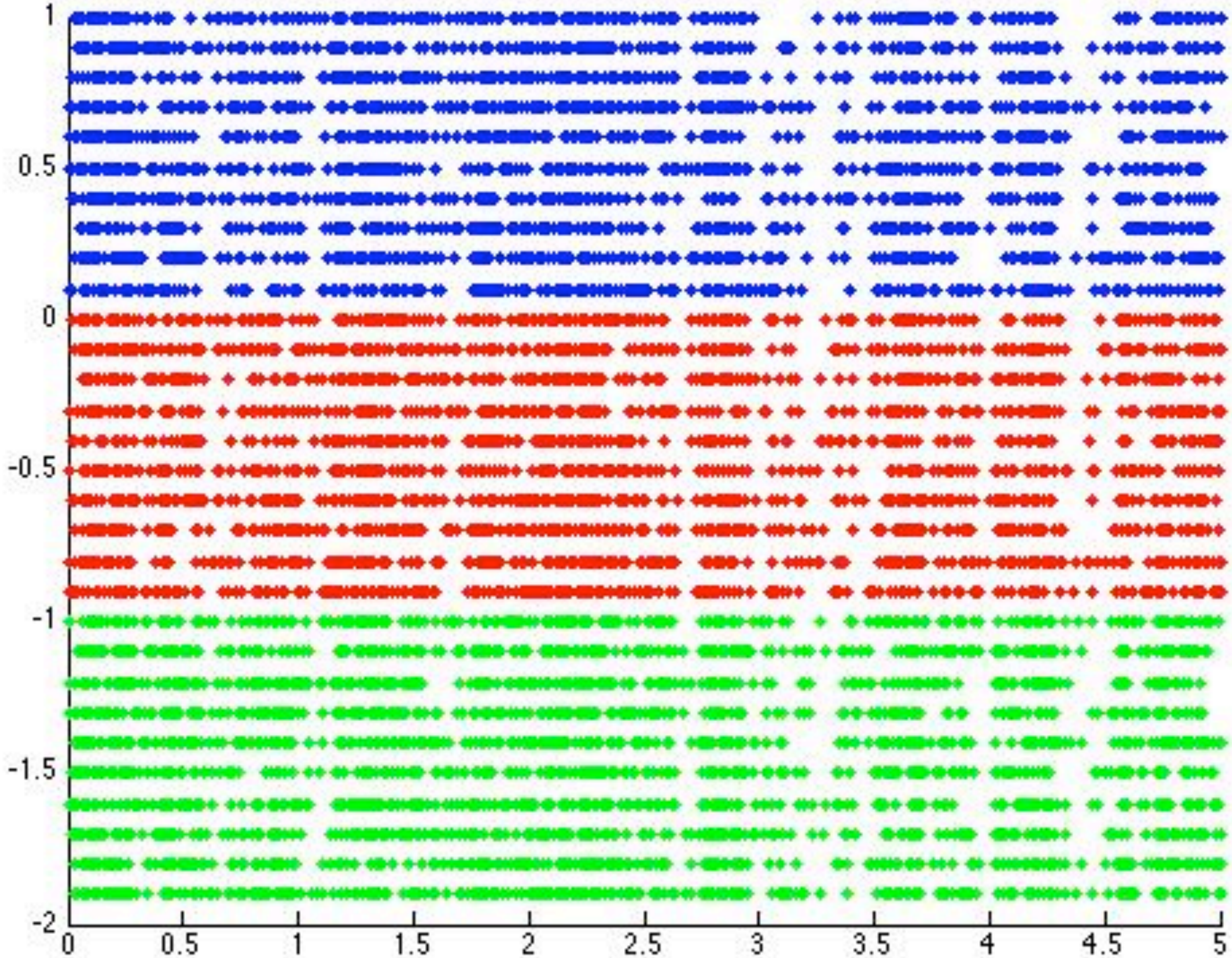
HMM

HMM



HMM
LNP
True
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
5
10
15
20
25


1
0.5
0
-0.5
-1
-1.5
-2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5