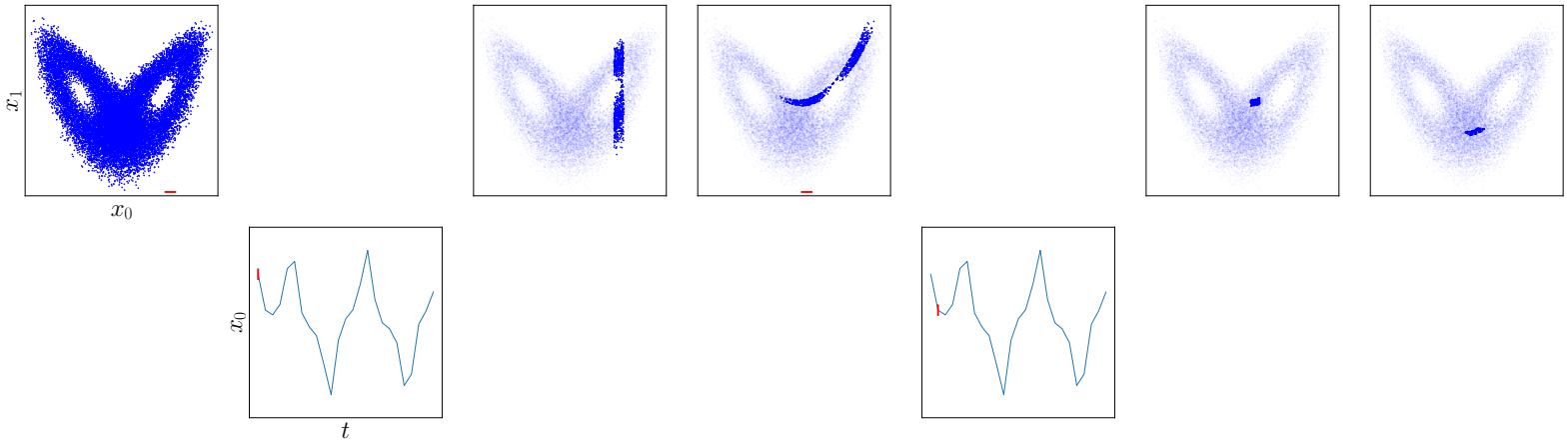
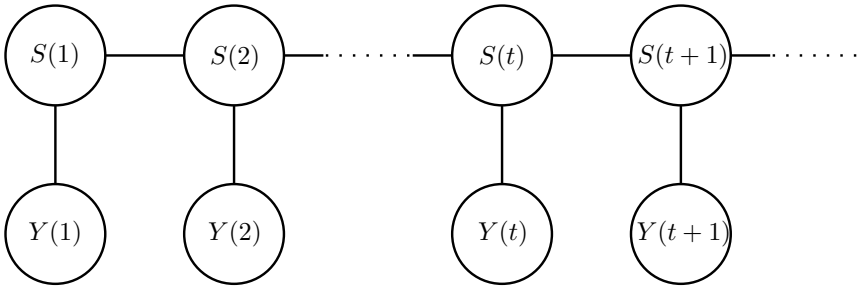
# Hidden Markov Models and Dynamical Systems

## State Space Perspective for Time Series[1]




## Models




The independence relations, or Bayes net, for a *state space model*.

**Prior:** $p_{s_1}$

**Dynamics:** $p_{s_{t+1}|s_t}$

**Observation** $p_{y_t|s_t}$

[1] Slides at fraserphysics.com/mlds20_doc.pdf

# Recursive Data Assimilation

$$
\begin{aligned}
p(s_1) & & \text{state prior} \\
p(s_1, y_1) &= p(y_1|s_1)p(s_1) & \\
p(y_1) &= \int p(s_1, y_1) ds_1 & \\
p(s_1|y_1) &= \frac{p(y_1, s_1)p(s_1)}{p(y_1)} & \text{observation update} \\
p(s_2|y_1) &= \int p(s_2|s_1)p(s_1|y_1) ds_1 & \text{state forecast}
\end{aligned}
$$

# Two Simple Models

Data assimilation was worked out exactly for two simple model classes in the 1960s:

**Linear Gaussian (Kalman Filter):**

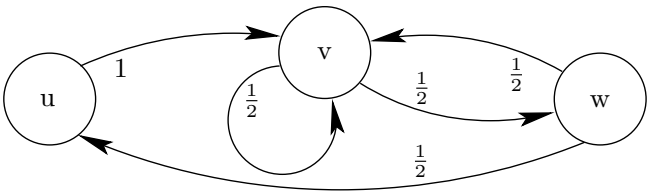
$$
\begin{aligned}
s_{t+1}|s_t &\sim \mathcal{N}(s_{t+1} - D \cdot s_t, \Sigma_S) & \text{state dynamics} \\
y_t|s_t &\sim \mathcal{N}(y_t - O \cdot s_t, \Sigma_Y) & \text{observation}
\end{aligned}
$$

**Discrete States and Discrete Observations:** (Hidden Markov Models)

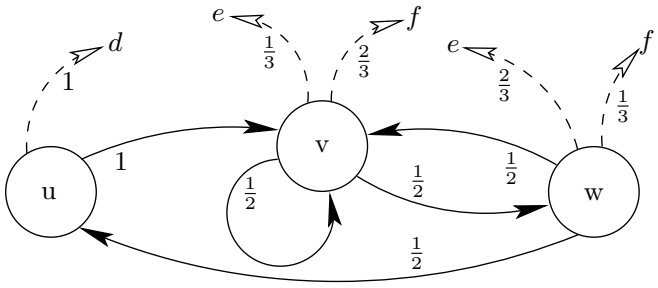
$\mathbf{p_{s_1}}$ A *vector* of prior probabilities for initial state

$\mathbf{p_{s_{t+1}|s_t}}$ A *matrix* of state transition probabilities

$\mathbf{p_{y_t|s_t}}$ A *matrix* of observation probabilities




A *Markov* model. Typical output: $\ldots vvwvvwuv \ldots$



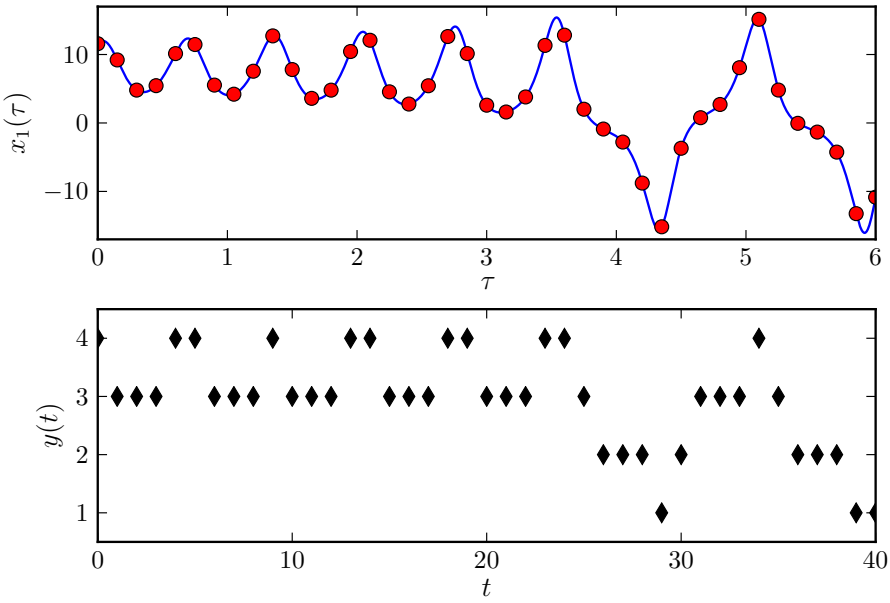

A *hidden Markov* model. Typical output: $\ldots effffdf \ldots$

**Q:** OK Boomer, why study these models from the '60s?

**A:** They are the hydrogen atom and simple harmonic oscillator of data assimilation. For some problems they are appropriate, and for more difficult problems they provide analogies for thinking about more sophisticated solutions or approximations.

# An Illustration

Partition of Lorentz state space




Continuous time $\tau$ and state $x$

Discrete time $t$ and observation $y$ (four levels)

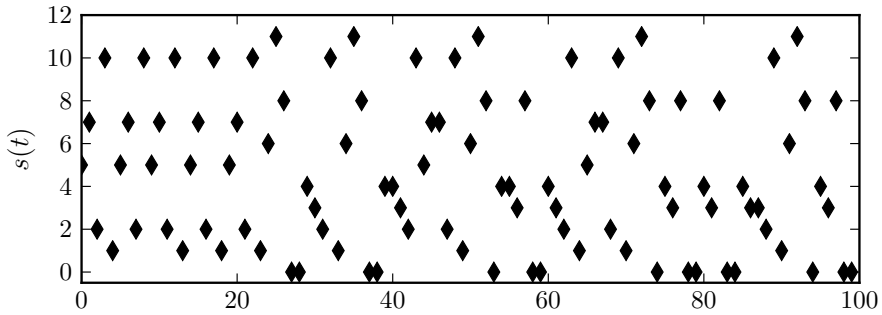
$$\theta \equiv \left\{p_{s_1},\ p_{s_{t+1}|s_t},\ p_{y_t|s_t}\right\}$$ The model parmeters

$$y_1^{20{,}000} \equiv [y_1, y_2, \ldots, y_{20{,}000}]$$ Training data

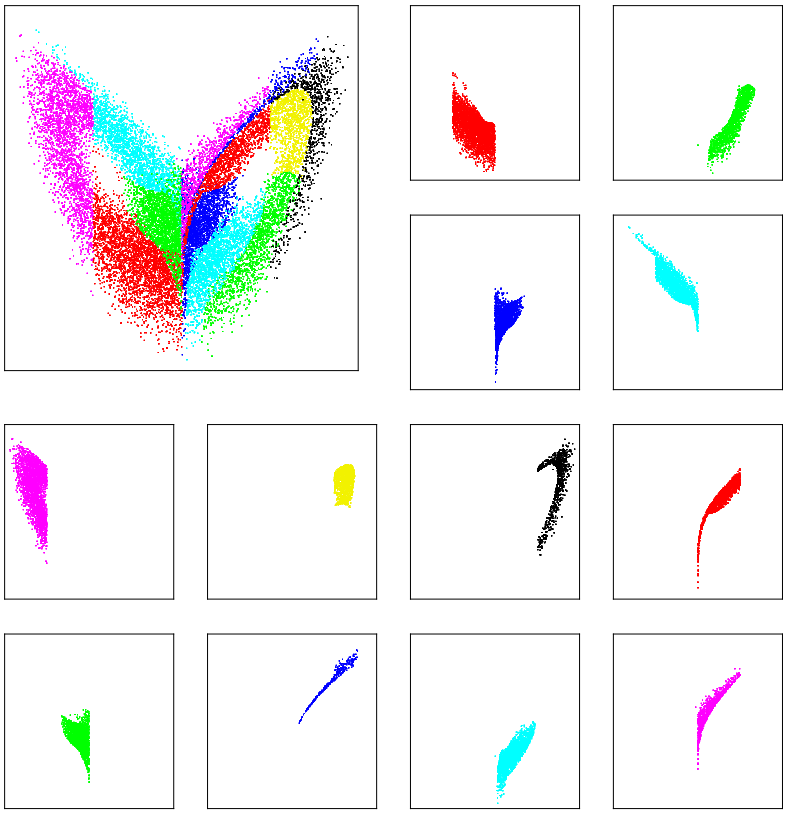
$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(y_1^{20{,}000}|\theta)$$ Maximum Likelihood Estimate

$$y_{20{,}001}^{40{,}000}$$ Testing data

$$\hat{c}_{20{,}001}^{40{,}000} = \underset{c_{20{,}001}^{40{,}000}}{\operatorname{argmax}}\, p\left(y_{20{,}001}^{40{,}000} \mid c_{20{,}001}^{40{,}000}\right)$$ Color by MLE given $\hat{\theta}$

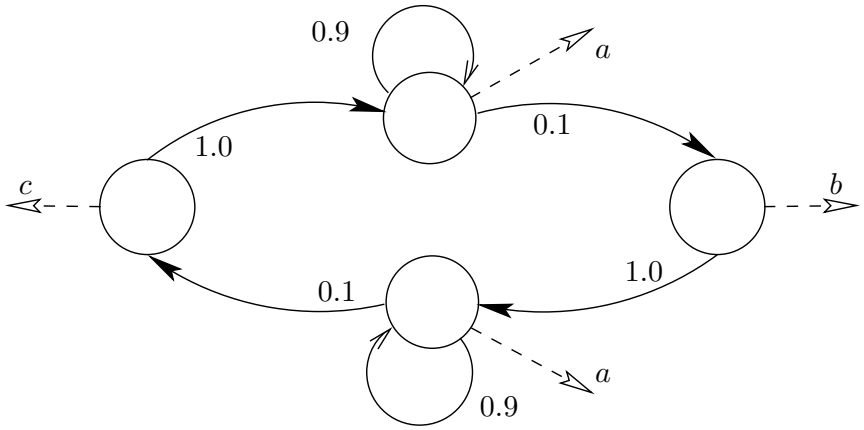



*Decoded* state sequence (12 possible states)



**Q:** Isn't an HMM just a high order Markov model?

**A:** No.




For a sequence $caaa \ldots aaa$ the probabilities for the next letter are $p_a = .9$ and $p_b = .1$ no matter how long the string of $a$s.

# Estimates

## Forward Filter

Given a sequence of observations $y_1^t$ and a model $\theta$, the *foreword* algorithm recursively calculates

$$\alpha(s_t, t) \equiv p(s_t|y_1^t),$$

the probability of each state at time $t$ given all observations up to time $t$:

$$\begin{aligned}
p\left(s_t|y_1^{t-1}\right) &= \sum_{s_{t-1}} p\left(s_t, s_{t-1}|y_{t-1}\right) && \text{state probability forecast} \\
p\left(s_t, s_{t-1}|y_{t-1}\right) &= p\left(s_s|s_{t-1}\right) p\left(s_{t-1}|y_{t-1}\right) \\
p\left(y_t|y_1^{t-1}\right) &= \sum_{s_t} p\left(s_t, y_t|y_1^{t-1}\right) && \text{observation probability forecast} \\
p\left(s_t, y_t|y_1^{t-1}\right) &= p\left(y_t|s_t\right) p\left(s_t|y_1^{t-1}\right) \\
p\left(s_t|y_t^t\right) &= \frac{p\left(s_t, y_t|y_1^{t-1}\right)}{p\left(y_t|y_1^{t-1}\right)} && \text{update state probability}
\end{aligned}$$

## Backward Filter and Smoothing

Similarly, given a sequence of observations $y_{t+1}^T$ and a model $\theta$, the *backward* algorithm recursively calculates the ratio

$$\beta(s_t, t) \equiv \frac{p\left(y_{t+1}^T|s_t\right)}{p\left(y_{t+1}^T|y_t^t\right)}.$$

The peculiar normalization of $\beta(s_t, t)$ prepares for calculating the *smoothed* probability of the states at any intermediate time $t : 1 \leq t \leq T$

$$p\left(s_t|y_1^T\right) = \alpha\left(s_t, t\right) \beta\left(s_t, t\right).$$

The analogous calculation for a linear Gaussian model, called *Kalman Smoothing*, combines the forward updated probability with the backward state forecast.

## Baum Welch Estimation

Given an initial model $\theta_1$ and observation data $y_1^T$, the Baum Welch algorithm (predates EM paper) converges to a local maximum of the likelihood

$$\hat{\theta} = \underset{\theta \in \text{neighborhood}}{\text{argmax}} \; p\left(y_1^T | \theta\right)$$

The algorithm alternates between calculating the conditional distribution of the unobserved states $p\left(s_1^T | y_1^T, \theta_n\right)$ and using that conditional distribution to reestimate parameters $\theta_{n+1}$. The key calculation is

$$p\left(s_{t+1}, s_t | y_1^T, \theta_n\right) \propto \beta\left(s_{t+1}, t+1\right) p\left(y_{t+1} | s_{t+1}, \theta_n\right) p\left(s_{t+1} | s_t, \theta_n\right) \alpha\left(s_t, t\right).$$

## Viterbi Algorithm

Given a sequence of observations $y_1^T$ and a model $\theta$, the Viterbi algorithm calculates the maximum likelihood sequence of states:

$$\hat{s_1^T} \equiv \underset{s_1^T}{\text{argmax}}\, p\left(y_1^T | s_1^T, \theta\right)$$

The sequence of maximum likelihood states is not in general the maximum likelihood sequence of states. In fact the sequence of maximum likelihood states may be an impossible sequence.

# Entropy

There are two notions of entropy (Shannon–McMillan–Breiman theorem says they are equal)

- The exponential rate at which the number of plausible sequences grows

$$H = \lim_{T\to\infty} \frac{1}{T} \log\left(\left|\left\{y_1^T\right\}_{\text{plausible}}\right|\right)$$

- The exponential rate at which the probability of each plausible sequence decays

$$H = \lim_{T\to\infty} \frac{-1}{T} \log\left(p\left(y_1^T\right)\right)$$

For a process of iid draws from from a discrete set $X$ with distribution $p$ one obtains the familiar

$$H = -\sum_x p(x) \log\left(p(x)\right).$$

For a chaotic process, there is a bound on how well one can predict future values. The remainder of this section explains how to calculate how close a prediction procedure or model is to that bound.

For a model, $\theta$, and a true $\phi$, the relative entropy is defined by the expectation

$$D(\phi||\theta) \equiv \mathbb{E}_\phi \log \frac{p_\phi(x)}{p_\theta(x)} = \mathbb{E}_\phi \log \frac{1}{p_\theta(x)} - H_\phi \equiv H_{\phi||\theta} - H_\phi \geq 0.$$

It is zero if and only if $\theta = \phi$ almost everywhere. If $\phi$ is an ergodic process, then

$$H_{\phi||\theta} = \lim_{T\to\infty} \frac{-1}{T} \log\left(p_\theta\left(y_1^T\right)\right)$$

almost everywhere.

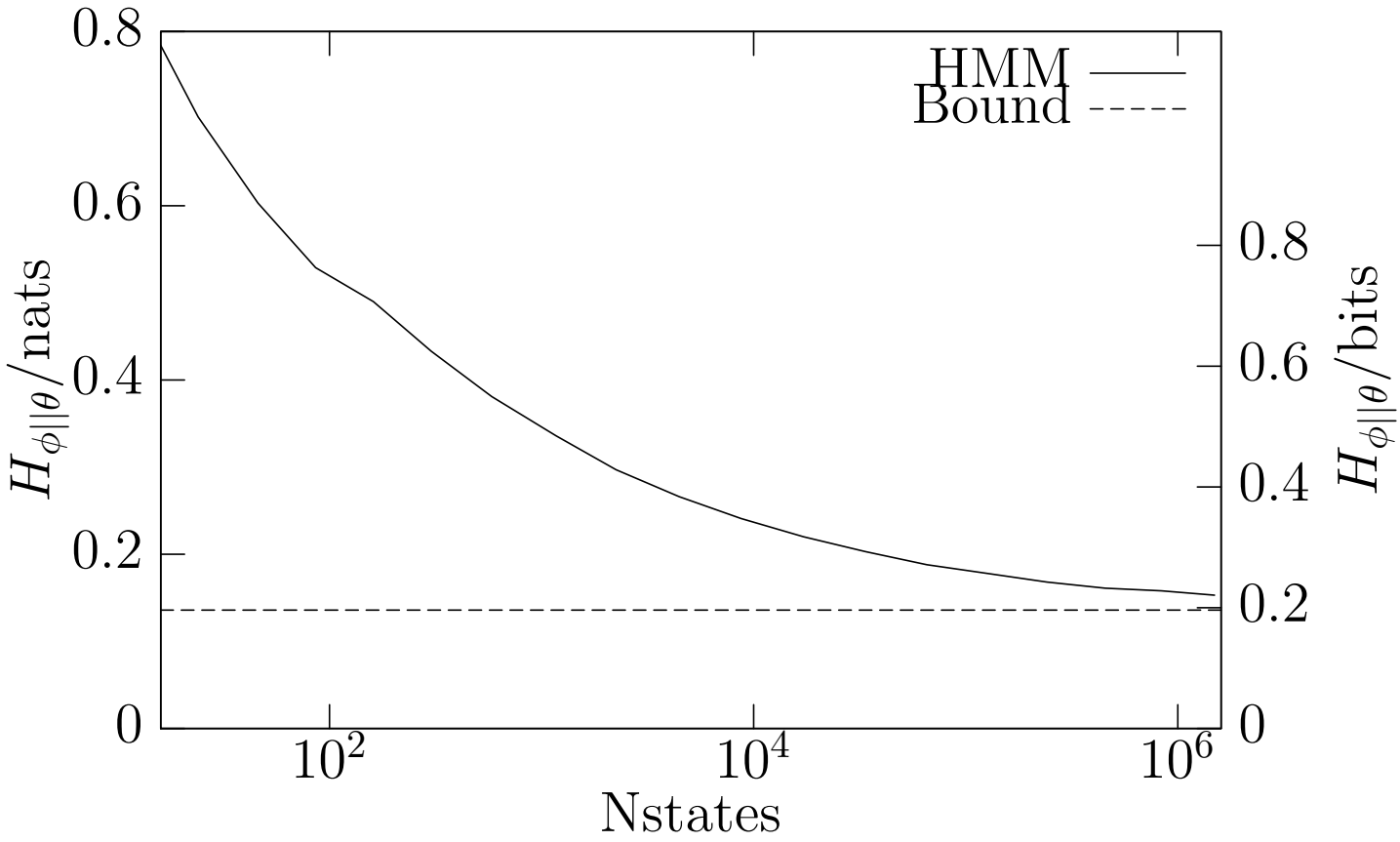
From numerical estimates of the Laypunov exponents, $\lambda_i$, of a chaotic dynamical system, $\phi$, one can estimate the entropy by

$$H_\phi = \sum_{i:\lambda_i>0} \lambda_i.$$

Thus the quantity

$$H_{\phi||\theta} - H_\phi = \frac{-1}{T} \log\left(p_\theta\left(y_1^T\right)\right) - \sum_{i:\lambda_i>0} \lambda_i$$

indicates how close the predictions of $\theta$ are to ideal predictions.


HMM
Bound
0.8
0.6
0.4
0.2
0
$H_{\phi||\theta}$/nats
$H_{\phi||\theta}$/bits
$10^2$
$10^4$
$10^6$
Nstates

# Conclusion[2]

Working with simple hidden Markov models provides connections to:

- Bayes nets
- Data assimilation
- Viterbi decoding (Dynamic programming)
- Kalman filtering
- EM algorithm
- Information theory
- Chaotic dynamical systems

[2]Slides at fraserphysics.com/mlds20_doc.pdf