Week 5, Lecture 10 - Hidden Markov Models

Aaron Meyer

Outline

- Administrative Issues
- Review
- Hidden Markov Models
- Applications

Based on slides from Aarti Singh.

Sequential Data

- So far we've largely assumed that we have independent data
- But what if we have sequences?
  - E.g. audio, a string of text, time-series data
  - Then our measurements are co-dependent
  - Assumption of independence often gives nonsensical results

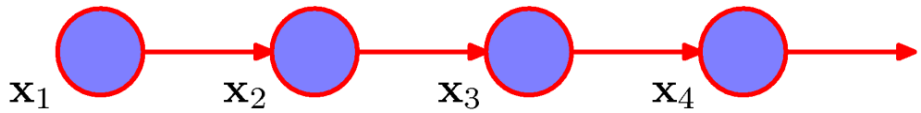
Markov Models

- Joint Distribution
  - $p(\mathbf{X}) = p(X_1, X_2, \ldots, X_n)$
  - $p(\mathbf{X}) = p(X_1)p(X_2 \mid X_1)p(X_3 \mid X_2, X_1) \ldots p(X_n \mid X_{n-1}, \ldots, X_1)$
  - $\prod_{i=1}^{n} p(X_n \mid X_{n-1}, \ldots, X_1)$
    - Chain rule
- Markov Assumption
  - $\prod_{i=1}^{n} p(X_n \mid X_{n-1}, \ldots, X_{n-m})$
    - Current observation only depends on past m observations

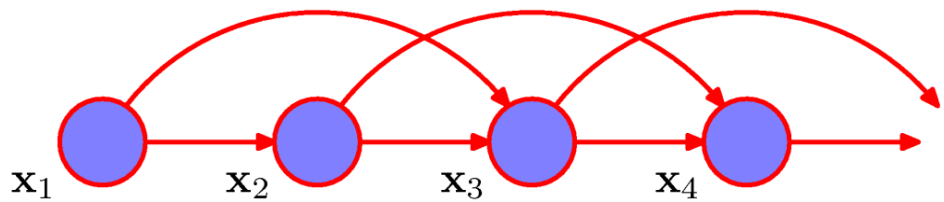
Markov Models

Markov Assumption

1st order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$$



2nd order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1}, X_{n-2})$$



Markov Models

Markov Assumption

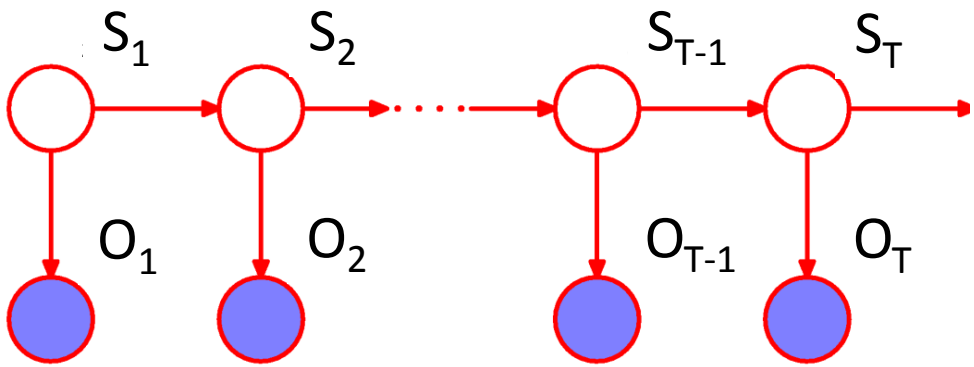
The number of parameters in a stationary model with K-ary variables

- 1st Order
  - $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n \mid X_{n-1})$
  - $O(K^2)$
- mth Order
  - $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n \mid X_{n-1}, \ldots, X_{n-m})$
  - $O(K^{m+1})$
- $n-1$th Order
  - $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n \mid X_{n-1}, \ldots, X_1)$
  - $O(K^n)$
  - No assumptions: Complete (but directed) graph

Homogeneous/stationary Markov model (probabilities don't depend on n)

Hidden Markov Models

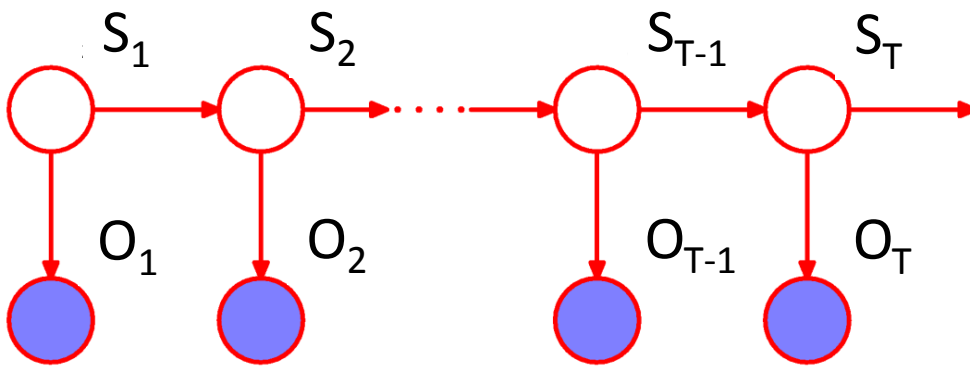
Distributions that characterize sequential data with few parameters but are not limited by strong Markov assumptions.



Observation space: $O_t \in \{y_1, \ldots, y_k\}$

Hidden states: $S_t \in \{1, \ldots, I\}$

Hidden Markov Models



$$p(S_1, \ldots, S_T, O_1, \ldots, O_T) = \prod_{t=1}^{T} p(O_t \mid S_t) \prod_{t=1}^{T} p(S_t \mid S_{t-1})$$

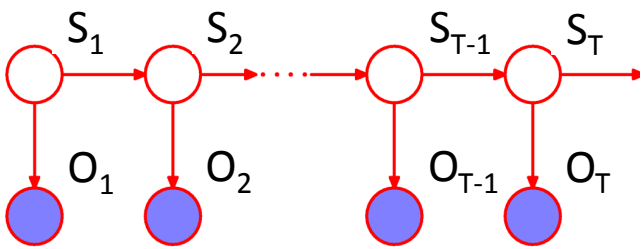
1st order Markov assumption on hidden states $\{S_t\}$ $t = 1, \ldots, T$ (can be extended to higher order).

Note: O_t depends on all previous observations $\{O_{t-1}, \ldots O_1\}$

Hidden Markov Models

Parameters for stationary/homogeneous Markov model

- Independent of T
- Initial probabilities: $p(S_1 = i) = \pi_i$
- Transition probabilities: $p(S_t = j \mid S_{t-1} = i) = p_{ij}$
- Emission probabilities: $p(O_t = y \mid S_t = i) = q_i^y$



$$p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = p(S_1) \prod_{t=2}^{T} p(S_t|S_{t-1}) \prod_{t=1}^{T} p(O_t|S_t)$$

HMM Example

The Dishonest Casino

A casino has two die:

- Fair dice
- Loaded dice:
  $P(6) = 1/2, P(1) = P(2) = P(3) = P(4) = P(5) = 1/10$

Casino player switches back & forth between fair and loaded die once every 20 turns.

HMM Problems

Given: A sequence of rolls by the casino player:

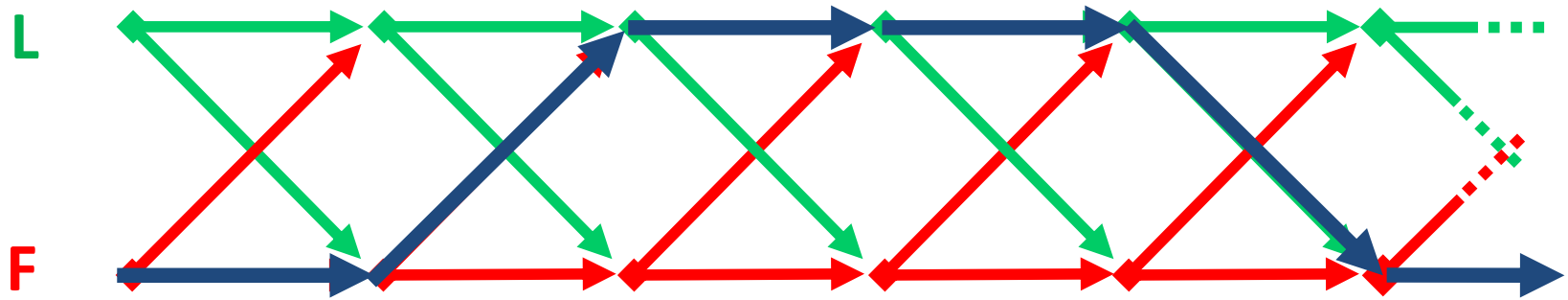
4352**6**1**6**2543**6**1**6**52**6**5342**6**5152**6**3**6**1524324132251543**6**

Questions:

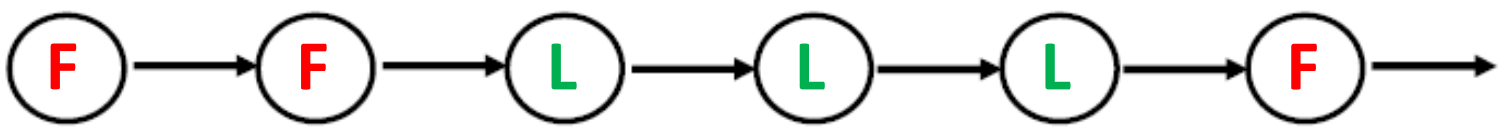
- How likely is this sequence, given our model of how the casino works?
  - This is the **evaluation** problem in HMMs
- What portion of the sequence was generated with the fair die, and what portion with the loaded die?
  - This is the **decoding** question in HMMs
- How biased is the loaded die? How fair is the fair die? How often does the casino player change from fair to loaded and back?
  - This is the **learning** question in HMMs

HMM Example

- Observed sequence: $\{O_t\}_{t=1}^{T}$

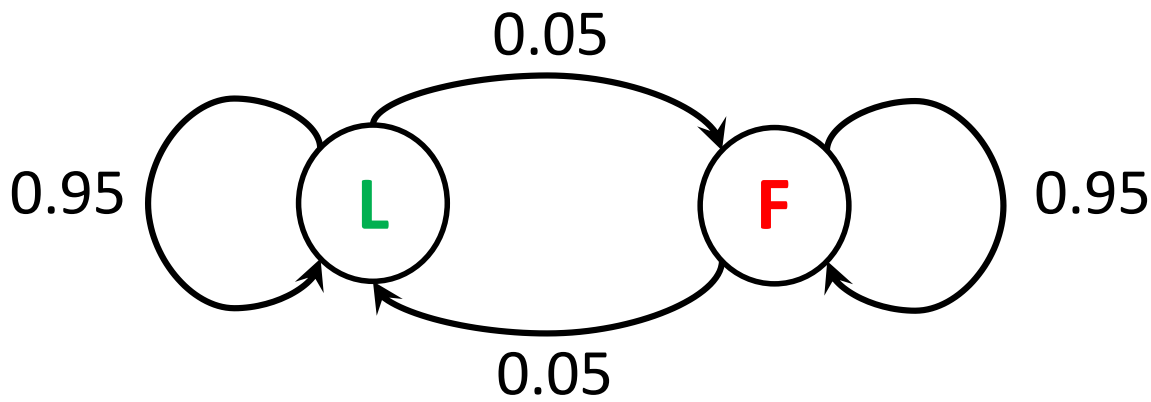


- Hidden sequence $\{S_t\}_{t=1}^{T}$ or segmentation):



State Space Representation

Switch between **F** and **L** once every 20 turns $(1/20 = 0.05)$



HMM Parameters

Initial probs	$P(S_1 = \mathbf{L}) = 0.5 = P(S_1 = \mathbf{F})$
Transition probs	$P(S_t = \mathbf{L}/\mathbf{F} \mid S_{t-1} = \mathbf{L}/\mathbf{F}) = 0.95$
	$P(S_t = \mathbf{F}/\mathbf{L} \mid S_{t-1} = \mathbf{L}/\mathbf{F}) = 0.05$
Emission probabilities	$P(O_t = y \mid S_t = \mathbf{F}) = 1/6 \quad y = 1,2,3,4,5,6$
	$P(O_t = y \mid S_t = \mathbf{L}) = 1/10 \quad y = 1,2,3,4,5$
	$= 1/2 \quad y = 6$

Three main problems in HMMs

- **Evaluation**: Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$
  - find the probability of observed sequence
- **Decoding**: Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$
  - find most probable sequence of hidden states
- **Learning**: Given HMM with unknown parameters and $\{O_t\}_{t=1}^T$ observation sequence
  - find parameters that maximize likelihood of observed data

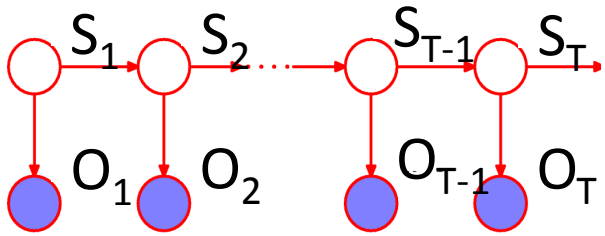
HMM Algorithms

- Evaluation
  - What is the probability of the observed sequence? **Forward Algorithm**
- Decoding
  - What is the probability that the third roll was loaded given the observed sequence? **Forward-Backward Algorithm**
  - What is the most likely die sequence given the observed sequence? **Viterbi Algorithm**
- Learning
  - Under what parameterization is the observed sequence most probable? **Baum-Welch Algorithm (EM)**

Evaluation Problem

Given HMM parameters $p(S_1), p(S_t \mid S_{t-1}), p(O_t \mid S_t)$ & observation sequence $\{O_t\}_{t=1}^T$

find probability of observed sequence



$$p(\{O_t\}_{t=1}^T) = \sum_{S_1,\ldots,S_T} p(\{O_t\}_{t=1}^T, \{S_t\}_{t=1}^T)$$

$$= \sum_{S_1,\ldots,S_T} p(S_1) \prod_{t=2}^T p(S_t|S_{t-1}) \prod_{t=1}^T p(O_t|S_t)$$

requires summing over all possible hidden state values at all times – K^T exponential # terms!

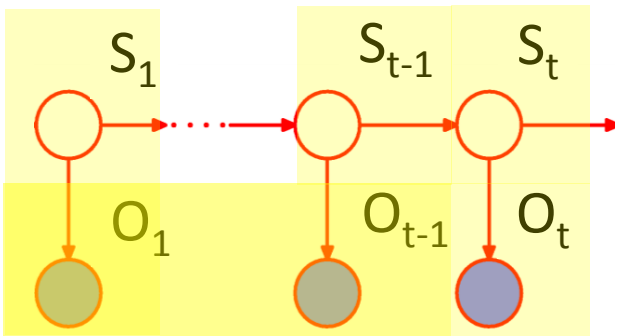
Instead: $p(\{O_t\}_{t=1}^T) = \sum_k \underbrace{p(\{O_t\}_{t=1}^T, S_T = k)}_{\alpha_T^k}$ Compute recursively

Forward Probability

$$p(\{O_t\}_{t=1}^T) = \sum_k p(\{O_t\}_{t=1}^T, S_T = k) = \sum_k \alpha_T^k$$

Compute forward probability α_t^k recursively over t

$$\alpha_t^k \quad := \quad p(O_1, \ldots, O_t, S_t = k)$$



Introduce S_{t-1}

.
.
.

Chain rule

Markov assumption

$$= \quad p(O_t|S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k|S_{t-1} = i)$$

Forward Algorithm

Can compute α_t^k for all k, t using dynamic programming:

- Initialize: $\alpha_1^k = p(O_1 \mid S_1 = k)p(S_1 = k) \ \ \forall k$
- Iterate: for $t = 2, \ldots, T$

$$\alpha_t^k = p(O_t \mid S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k \mid S_{t-1} = i) \ \ \forall k$$

- Termination:

$$p\left(\{O_t\}_{t=1}^T\right) = \sum_k \alpha_T^k$$

Decoding Problem 1

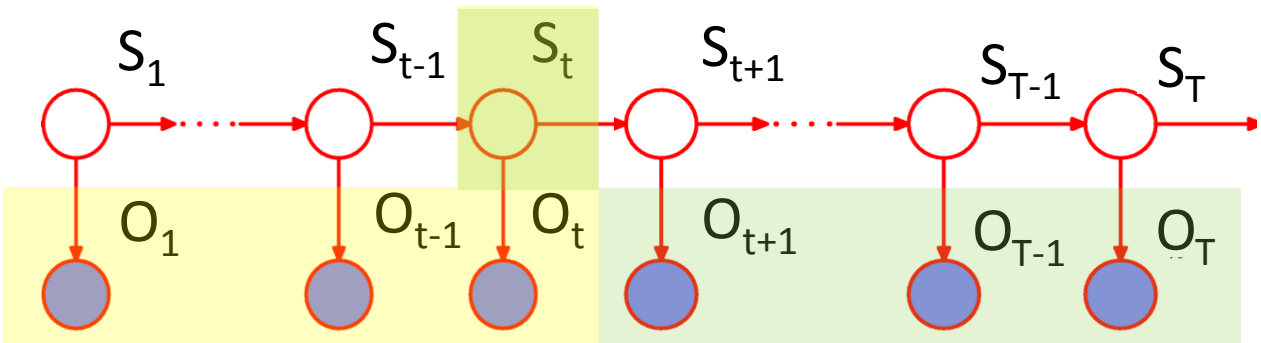
Given HMM parameters $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ & observation sequence $\{O_t\}_{t=1}^T$

find probability that hidden state at time t was k $p(S_t = k|\{O_t\}_{t=1}^T)$

$$
\begin{aligned}
p(S_t = k, \{O_t\}_{t=1}^T) &= p(O_1, \ldots, O_t, S_t = k, O_{t+1}, \ldots, O_T) \\
&= p(O_1, \ldots, O_t, S_t = k) p(O_{t+1}, \ldots, O_T | S_t = k)
\end{aligned}
$$

Compute recursively

α_t^k β_t^k

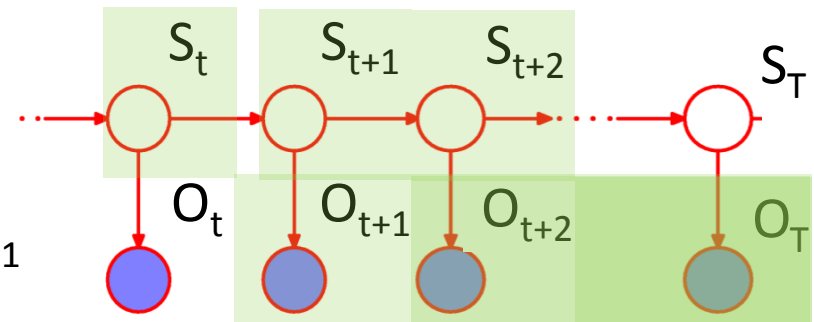


Backward Probability

$$p(S_t = k, \{O_t\}_{t=1}^T) = p(O_1, \ldots, O_t, S_t = k) p(O_{t+1}, \ldots, O_T | S_t = k) = \alpha_t^k \beta_t^k$$

Compute forward probability β_t^k recursively over t

$$\beta_t^k := p(O_{t+1}, \ldots, O_T | S_t = k)$$



Introduce S_{t+1}

.
.
.

Chain rule

Markov assumption

$$= \sum_i p(S_{t+1} = i | S_t = k) p(O_{t+1} | S_{t+1} = i) \beta_{t+1}^i$$

Backward Algorithm

Can compute $\beta_t^k \ \ \forall k, t$ using dynamic programming:

- Initialize: $\beta_T^k = 1 \ \ \forall k$
- Iterate: for t = T-1, . . . , 1

$$\beta_t^k = \sum_i p(S_{t+1} = i \mid S_t = k) p(O_{t+1} \mid S_{t+1} = i) \beta_{t+1}^i \ \ \forall k$$

- Termination: $p(S_t = k, \{O_t\}_{t=1}^T) = \alpha_t^k \beta_t^k$

$$p(S_t = k \mid \{O_t\}_{t=1}^T) = \frac{p(S_t = k, \{O_t\}_{t=1}^T)}{p(\{O_t\}_{t=1}^T)} = \frac{\alpha_t^k \beta_t^k}{\sum_i \alpha_t^i \beta_t^i}$$

Most likely state vs. Most likely sequence

- Most likely state assignment at time t

$$\arg\max_k p(S_t = k \mid \{O_t\}_{t=1}^T) = \arg\max_k \alpha_t^k \beta_t^k$$

  - E.g. Which die was most likely used by the casino in the third roll given the observed sequence?

- Most likely assignment of state sequence

$$\arg\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T \mid \{O_t\}_{t=1}^T)$$

  - E.g. What was the most likely sequence of die rolls used by the casino given the observed sequence?

Not the same!

Decoding Problem 2

Given HMM parameters $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ & observation sequence $\{O_t\}_{t=1}^T$

find most likely assignment of state sequence

$$\arg \max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T) = \arg \max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T)$$

$$= \arg \max_k \underbrace{\max_{\{S_t\}_{t=1}^{T-1}} p(S_T = k, \{S_t\}_{t=1}^{T-1}, \{O_t\}_{t=1}^T)}_{V_T^k}$$

Compute recursively

V_T^k - probability of most likely sequence of states ending at state $S_T = k$

Viterbi Decoding

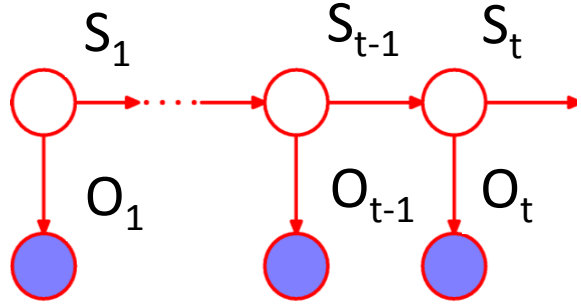
$$\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k V_T^k$$

Compute probability V_t^k recursively over t

$$V_t^k := \max_{S_1,\ldots,S_{t-1}} p(S_t = k, S_1, \ldots, S_{t-1}, O_1, \ldots, O_t)$$

. Bayes rule

. Markov assumption



$$= p(O_t|S_t = k) \max_i p(S_t = k|S_{t-1} = i) V_{t-1}^i$$

Viterbi Algorithm

Can compute $V_t^k \ \ \forall k, t$ using dynamic programming:

- Initialize: $V_1^k = p(O_1|S_1=k)p(S_1 = k)$ for all k
- Iterate: for t = 2, …, T

$$V_t^k = p(O_t|S_t = k) \max_i p(S_t = k|S_{t-1} = i)V_{t-1}^i \quad \text{for all k}$$

- Termination: $\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k V_T^k$

Traceback:

$$S_T^* = \arg\max_k V_T^k$$

$$S_{t-1}^* = \arg\max_i p(S_t^*|S_{t-1} = i)V_{t-1}^i$$

Computational complexity

What is the running time for Forward, Forward-Backward, Viterbi?

$$\alpha_t^k = q_k^{O_t} \sum_i \alpha_{t-1}^i p_{i,k}$$

$$\beta_t^k = \sum_i p_{k,i} q_i^{O_{t+1}} \beta_{t+1}^i$$

$$V_t^k = q_k^{O_t} \max_i p_{i,k} V_{t-1}^i$$

$O(K^2T)$ linear in T instead of $O(K^T)$ exponential in T!

Learning Problem

- Given HMM with:
  - unknown parameters $\theta = \{\{\pi_i\}, \{p_{ij}\}, \{q_i^k\}\}$
  - observation sequence $O = \{O_t\}_{t=1}^T$
- Find parameters that maximize likelihood of observed data
  - $\arg\max_\theta p\left(\{O_t\}_{t=1}^T \mid \theta\right)$
  - But likelihood doesn't factorize since observations not i.i.d.
  - hidden variables: state sequence
- EM (Baum-Welch) Algorithm:
  - E-step: Fix parameters, find expected state assignments
  - M-step: Fix expected state assignments, update parameters

Baum-Welch (EM) Algorithm

- **E-step** – Fix parameters, find expected state assignments

$$\gamma_i(t) = p(S_t = i|O,\theta) = \frac{\alpha_t^i \beta_t^i}{\sum_j \alpha_t^j \beta_t^j}$$

Forward-Backward algorithm

$$\xi_{ij}(t) = p(S_{t-1}= i, S_t = j|O,\theta)$$

$$= \frac{p(S_{t-1} = i|O,\theta)p(S_t = j, O_t,\ldots,O_T|S_{t-1} = i,\theta)}{p(O_t,\ldots,O_T|S_{t-1} = i,\theta)}$$

$$= \frac{\gamma_i(t-1)\ p_{ij}\ q_j^{O_t}\ \beta_t^j}{\beta_{t-1}^i}$$

Baum-Welch (EM) Algorithm

- Start with random initialization of parameters

- **E-step**

$$\gamma_i(t) = p(S_t = i|O, \theta)$$

$$\xi_{ij}(t) = p(S_{t-1}= i, S_t = j|O, \theta)$$

$\sum_{t=1}^{T} \gamma_i(t)$ = expected # times in state i

$\sum_{t=1}^{T\text{-}1} \gamma_i(t)$ = expected # transitions from state i

$\sum_{t=1}^{T-1} \xi_{ij}(t)$ = expected # transitions from state i to j

- **M-step**

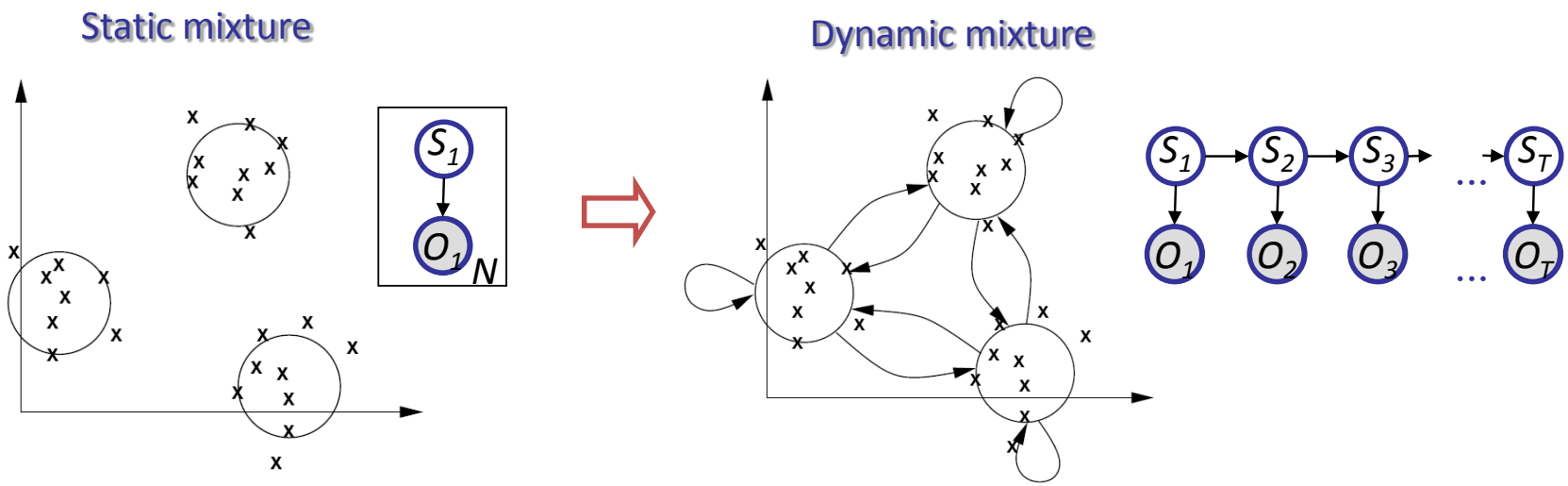
$$\pi_i = \gamma_i(1)$$

$$p_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$q_i^k = \frac{\sum_{t=1}^{T} \delta_{O_t=k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Some connections

- HMM & Dynamic Mixture Models
  - $p(O_t) = \sum_{S_t} p(O_t \mid S_t) p(S_t)$
- Choice of mixture component depends on choice of components for previous observations



Connections to Other Models

ODEs

- If we average over many individuals following a Markov process, we get a system of ODEs
- Multiple Markov models can give rise to the same ODEs due to transition probabilities

Latent variable models

- Many techniques we've talked about separate the data from the process generating it

HMMs: What You Should Know

- Useful for modeling sequential data with few parameters using discrete hidden states that satisfy Markov assumption
- Representation
  - Initial probability
  - Transition probabilities
  - Emission probabilities
- Algorithms for inference and learning in HMMs
  - Computing marginal likelihood of the observed sequence: *forward algorithm*
  - Predicting a single hidden state: forward-backward
  - Predicting an entire sequence of hidden states: Viterbi
  - Learning HMM parameters: an EM algorithm known as Baum-Welch

Example: Cancer Stem Cells

Stochastic State Transitions Give Rise to Phenotypic Equilibrium in Populations of Cancer Cells

Piyush B. Gupta,[1,6,*] Christine M. Fillmore,[2] Guozhi Jiang,[1] Sagi D. Shapira,[1] Kai Tao,[3] Charlotte Kuperwasser,[2,3] and Eric S. Lander[1,4,5,*]

[1]Broad Institute, Cambridge, MA 02142, USA
[2]Department of Anatomy and Cellular Biology, Sackler School of Graduate Biomedical Sciences, Tufts University School of Medicine, 136 Harrison Avenue, Boston, MA 02111, USA
[3]Molecular Oncology Research Institute, Tufts Medical Center, Boston, MA 02111, USA
[4]Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02142, USA
[5]Department of Systems Biology, Harvard Medical School, Boston, MA 02115, USA
[6]Present address: Department of Biology, Massachusetts Institute of Technology, and Whitehead Institute for Biomedical Research, Cambridge, MA 02142, USA
*Correspondence: pgupta@wi.mit.edu (P.B.G.), lander@broadinstitute.org (E.S.L.)
DOI 10.1016/j.cell.2011.07.026

SUMMARY

Cancer cells within individual tumors often exist in distinct phenotypic states that differ in functional attributes. While cancer cell populations typically display distinctive equilibria in the proportion of cells in various states, the mechanisms by which this occurs are poorly understood. Here, we study the

Example: Cancer Stem Cells

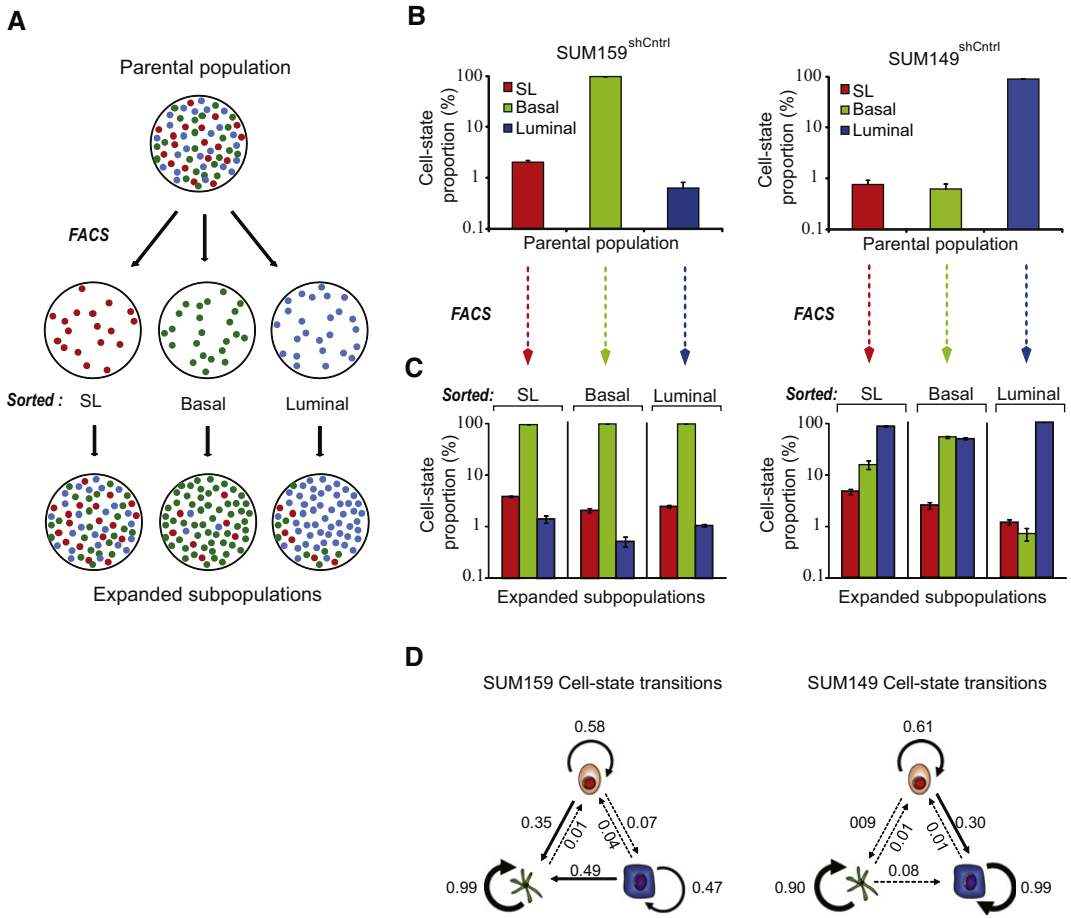


Figure 2. Determination of Breast Cancer Cell-State Transition Probabilities from Population Cell-State Proportions
(A) Schematic of experimental procedure used to determine cell-state transition dynamics.
(B) Proportions of cell-states in parental SUM159^{shCntrl} and SUM149^{shCntrl} breast cancer lines.
(C) Cellular subpopulations in stem-like (SL), basal, or luminal states were isolated by FACS with antibodies directed against the CD44, CD24, and EpCAM cell-surface antigens. Bar charts show the proportion of cells in each cell-differentiation state as assessed by FACS after in vitro culture for 6 days.
(D) Lineage hierarchies for the SUM159^{shCntrl} and SUM149^{shCntrl} lines were calculated from the data in (C). The corresponding cell-state transition probabilities for each cell line are shown. Solid arrows denote transition probabilities greater than 0.1. Dashed arrows denote transition probabilities between 0.01 and 0.1.
See also Figure S1.

Example: Cancer Stem Cells

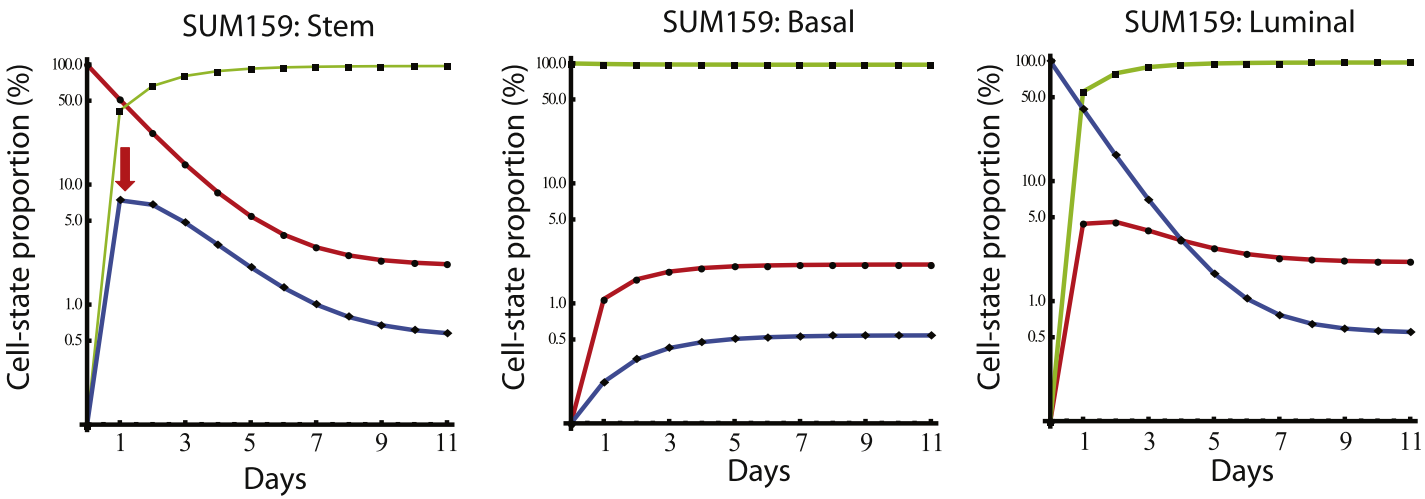


Figure 3. Prediction of Cell-State Dynamics with the Stochastic Cell-State Transition Model
The transition probabilities determined for the SUM159 line were used to compute the expected cell-state proportions over time for isolated subpopulations of stem-like, basal, or luminal cells. For all three isolated subpopulations, an eventual return to equilibrium cell-state proportions is predicted. The model predicts a transient increase in the proportion of SUM159 luminal cells one day after isolation of a stem-like subpopulation (red arrow).

Example: Cancer Stem Cells

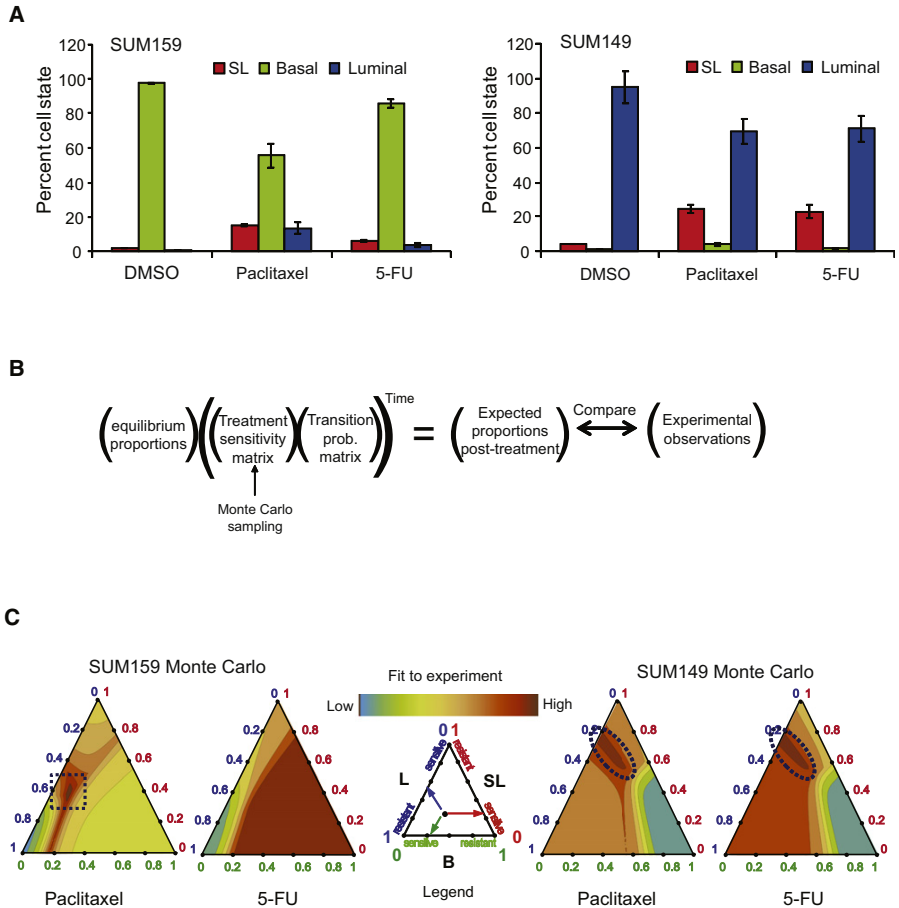


Figure 4. Analysis of Chemical Treatment Sensitivities by Monte Carlo Simulation with the Stochastic Cell-State Transition Model

(A) Cell-state proportions after a 6 day treatment with either paclitaxel or 5-FU are shown for SUM159 and SUM149 populations.

(B) A schematic showing how differential chemical treatment sensitivities can be incorporated into the Markov stochastic model. The matrix of transition probabilities is premultiplied by a differential viability matrix. The diagonal entries of this matrix, (v_B, v_S, and v_L), encode the survival probabilities of basal, stem-like and luminal cells in the presence of chemical treatment. These entries are normalized to sum to 1. Monte Carlo simulation is performed by random sampling of differential viability vectors.

(C) The results of Monte Carlo simulation, shown as ternary density contour plots in which coloration represents the normalized difference (see the Experimental Procedures for the metric used) between experimentally observed and predicted cell-state proportions following chemical treatment. Ten thousand random points within the triangle simplex were sampled, each representing a distinct choice of differential viability vector, (v_B, v_S, v_L), with entries ranging from zero (sensitive) to one (resistant).

Example: Cancer Stem Cells

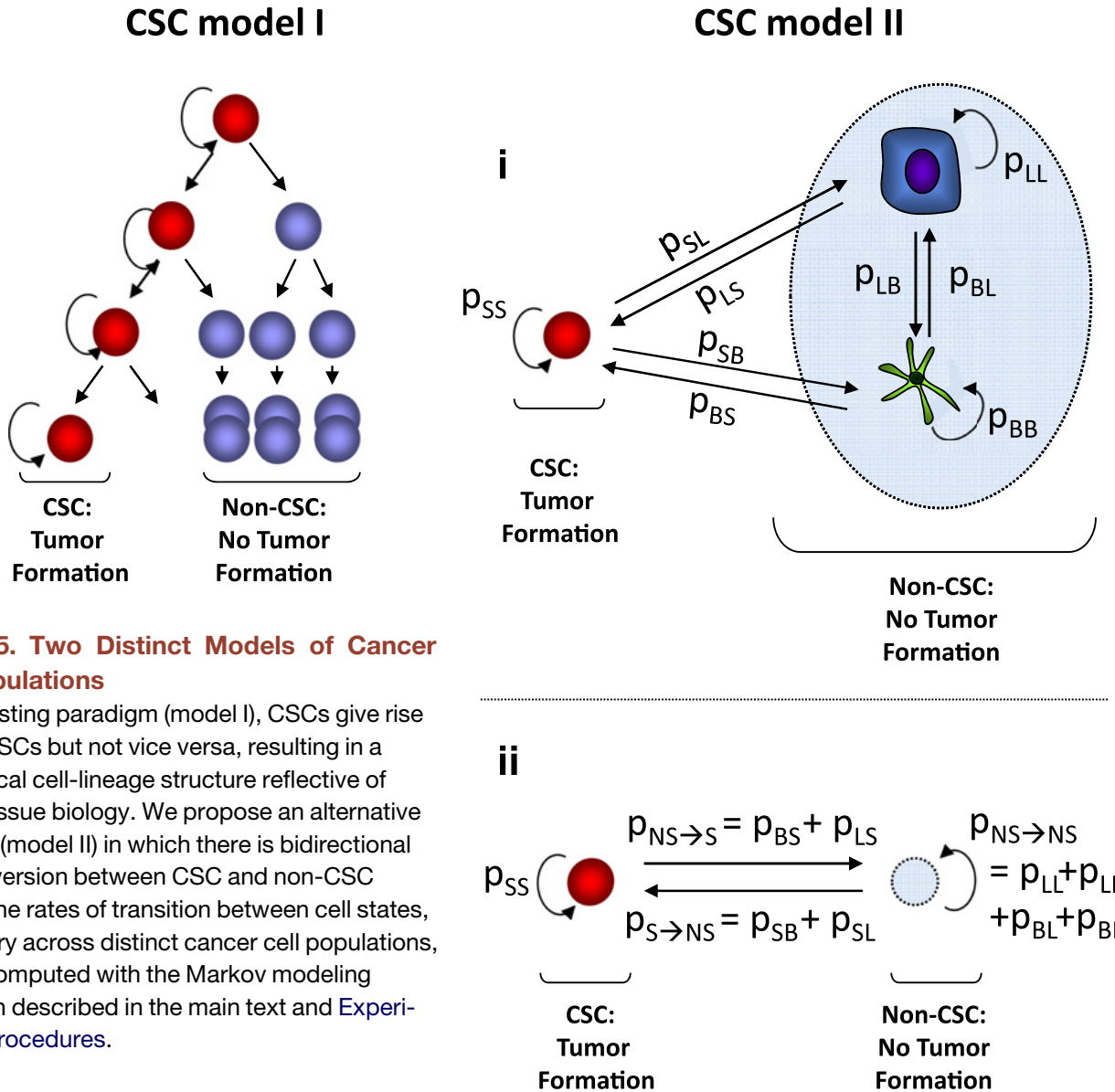


Figure 5. Two Distinct Models of Cancer Cell Populations

In the existing paradigm (model I), CSCs give rise to non-CSCs but not vice versa, resulting in a hierarchical cell-lineage structure reflective of normal tissue biology. We propose an alternative scenario (model II) in which there is bidirectional interconversion between CSC and non-CSC states. The rates of transition between cell states, which vary across distinct cancer cell populations, can be computed with the Markov modeling approach described in the main text and Experimental Procedures.

Example 2: Sequence Prediction

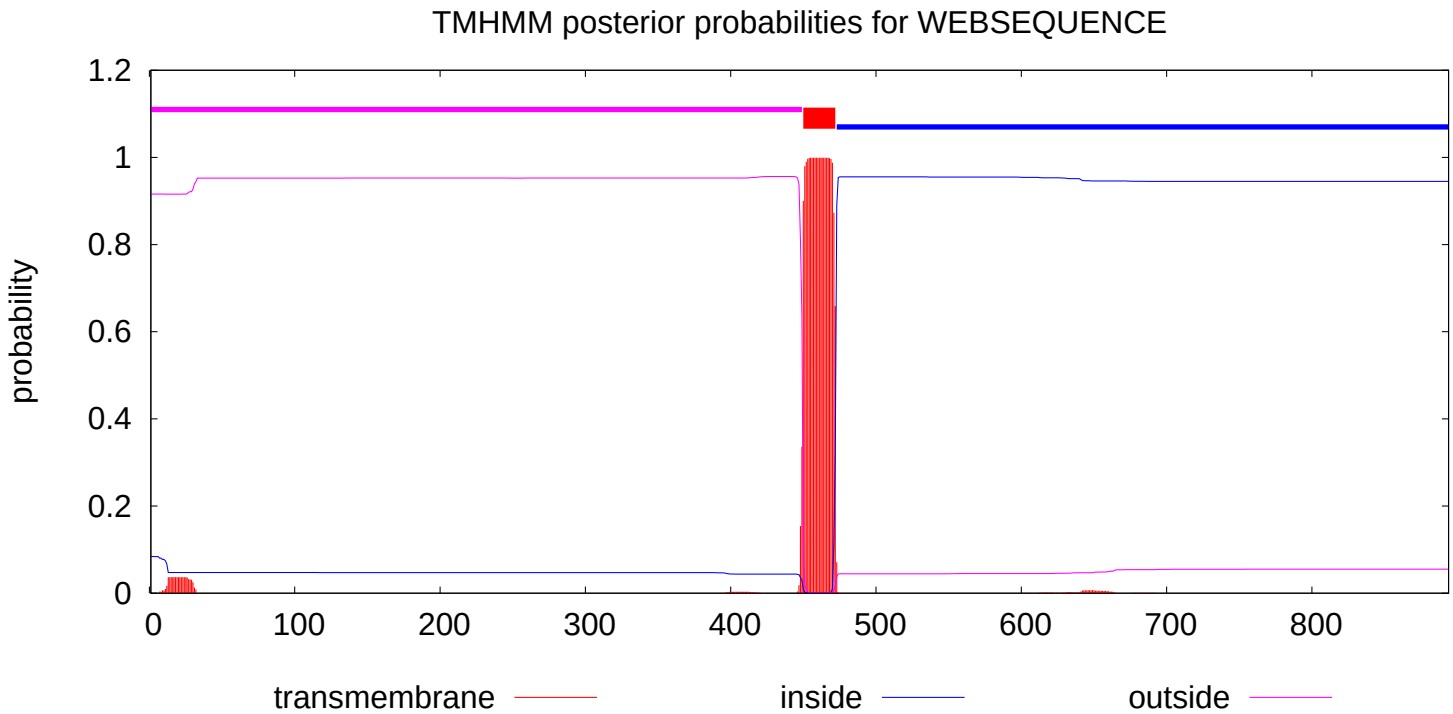


Figure: AXL prediction

Implementation

```
from hmmlearn.hmm import GaussianHMM

# Make an HMM instance and execute fit
model = GaussianHMM(n_components=4,
                    covariance_type="diag",
                    n_iter=1000).fit(X)

# Predict the optimal sequence of internal hidden state
hidden_states = model.predict(X)

print("Transition matrix")
print(model.transmat_)

print("Means and vars of each hidden state")
for i in range(model.n_components):
    print("{0}th hidden state".format(i))
    print("mean = ", model.means_[i])
    print("var = ", np.diag(model.covars_[i]))
```

Implementation

Find most likely state sequence corresponding to `X`.

```
model.decode(X, algorithm=None)
```

algorithm: Decoder algorithm. Must be one of “viterbi” or “map”. “viterbi” default.

Provides `logprob` and `state_sequence`.

Further Reading

- `hmmlearn`
- What is a hidden Markov model?
- Linear Digressions - Hidden Markov Models (part 2)