Sequential Model

Semester 1, 2021

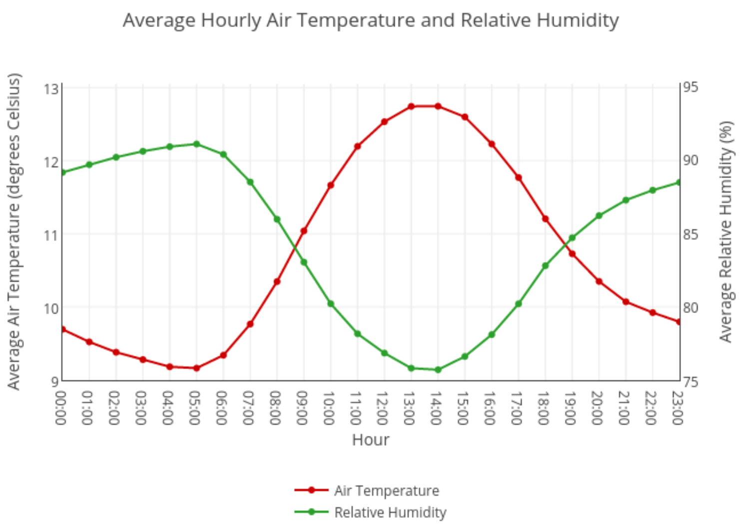
Ling Luo

Outline

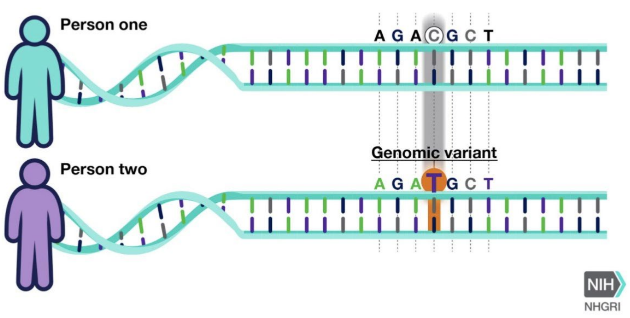
- Introduction
- Hidden Markov Models
- Applications

Structured Classification

- To date, we have always considered each instance independently, but in many tasks, there is “structure” between instances
  - **Sequential structure**: time series analysis, speech recognition, genomic data

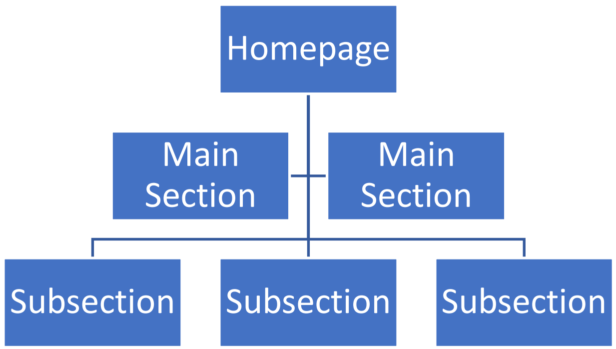


Source: https://chart-studio.plotly.com/~chloecrossman/152/

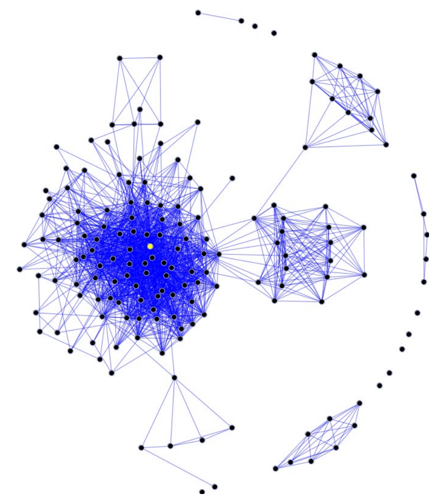


Source: https://www.genome.gov/Health/Genomics-and-Medicine/Polygenic-risk-scores

Structured Classification



Hierarchical structure



Social network graph

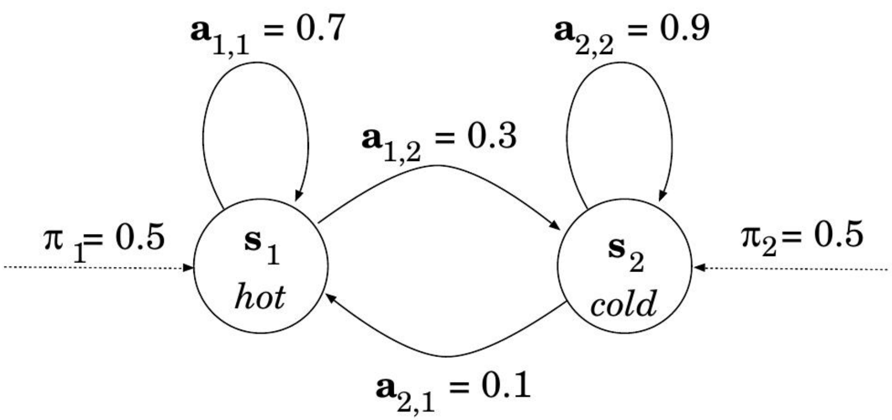
- **Hierarchical structure**: classifying web pages within a website
- **Graph structure**: deriving an influence matrix for a social network

- This calls for **structured classification** models which are able to capture the interaction between instances

Figure source (right): https://commons.wikimedia.org/w/index.php?curid=6057981

Markov Chains

- A Markov chain describes the system that transits from one state to another according to certain probabilistic rules
  - **States**: a set $S = \{s_i\}$
  - **Initial state distribution**: $\Pi = \{\pi_i\}, \sum_i \pi_i = 1$
  - **Transition probability matrix**: $A = \{a_{ij}\}, \sum_j a_{ij} = 1$ for $\forall i$



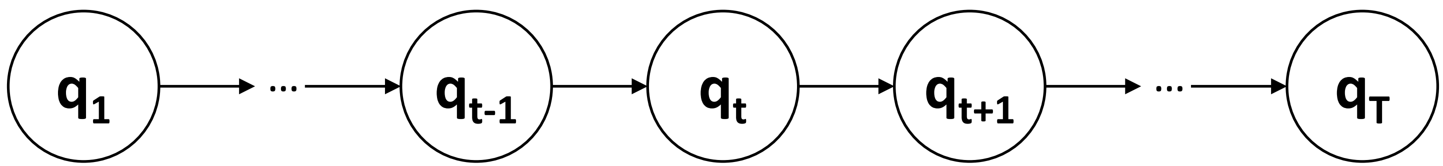
$$S = \{\text{hot}, \text{cold}\}$$

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix} \qquad \pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Markov Chains

Markov chains assumption

- For a sequence of states $q_1, q_2, \dots, q_{T-1}, q_T$ at steps $t = 1,2, \dots, T$

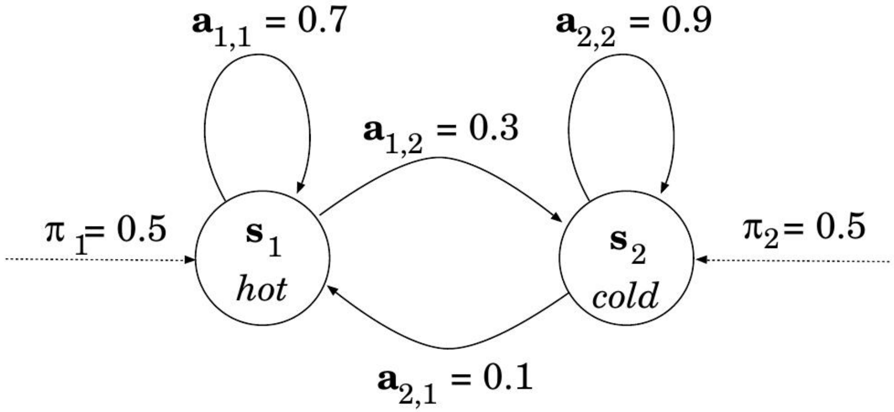


- State q_t *only depends* on the immediately preceding state q_{t-1}

$$P(q_t | q_1, \dots, q_{t-1}) = P(q_t | q_{t-1})$$

Markov Chains

- What is the probability of observing *hot, hot, cold*?



$$S = \{\text{hot}, \text{cold}\}$$

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix} \qquad \pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$P(q_1 = hot, q_2 = hot, q_3 = cold)$$

$$= P(q_3 = cold | q_2 = hot, q_1 = hot) P(q_2 = hot, q_1 = hot)$$

$$= P(q_3 = cold | q_2 = hot) P(q_2 = hot | q_1 = hot) P(q_1 = hot)$$

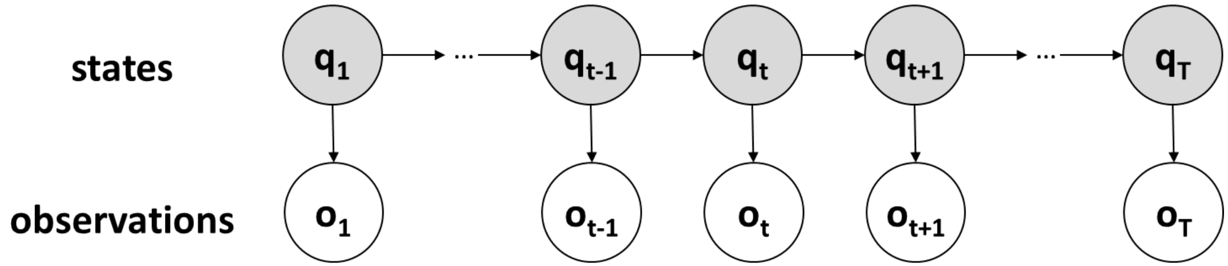
$$= 0.3 \times 0.7 \times 0.5 = 0.105$$

Hidden Markov Models

- Modelling
- Evaluation
- Decoding
- Learning

Hidden Markov Models

- We can see a sequence of observations, but the sequence of states is hidden

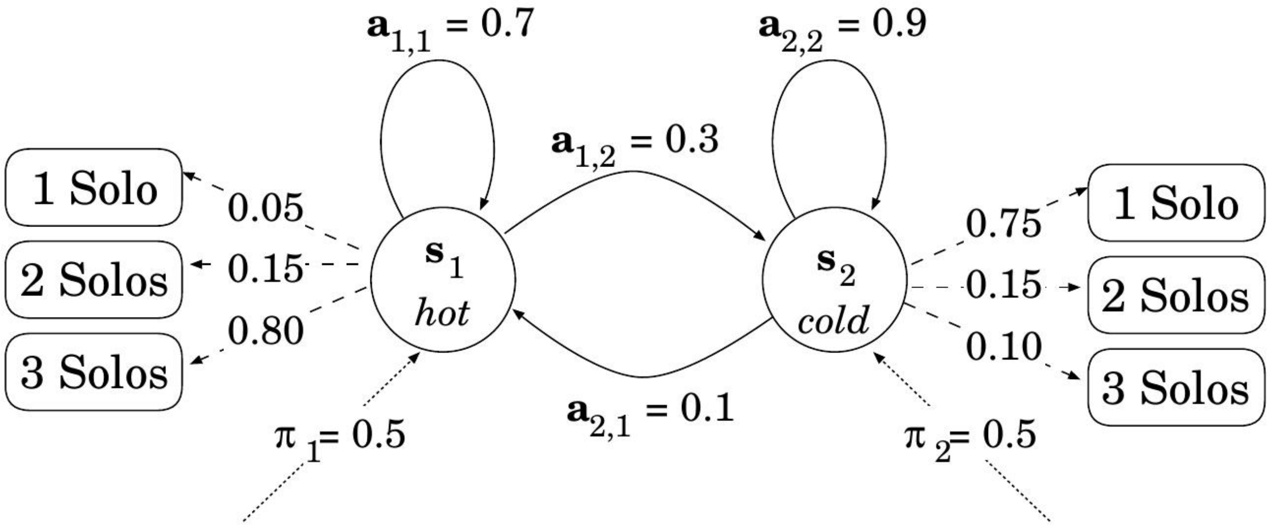


- Hidden Markov Models (HMM): $\mu = (A, B, \Pi)$
  - **States**: a set $S = \{s_i\}$
  - **Observations**: a set $\mathrm{O} = \{o_k\}$
  - **Initial state distribution**: $\Pi = \{\pi_i\}, \sum_i \pi_i = 1$
  - **Transition probability matrix**: $\mathrm{A} = \{a_{ij}\}, \sum_j a_{ij} = 1 \text{ for } \forall i$
  - **Output probability matrix**: $\mathrm{B} = \{b_i(o_k)\}, \sum_k b_i(o_k) = 1 \text{ for } \forall i$

Hidden Markov Models

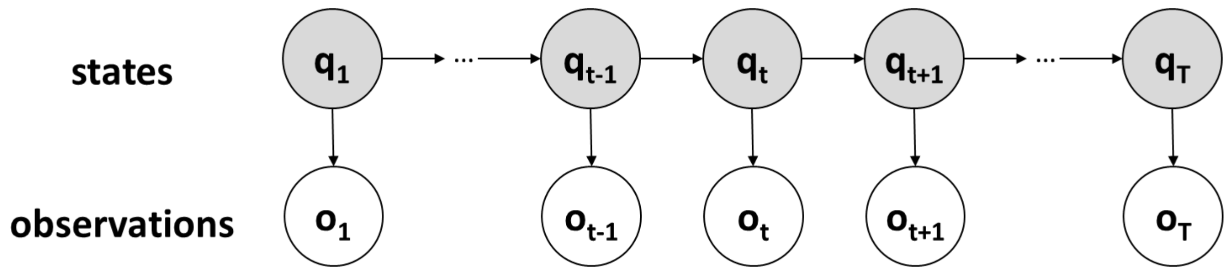
$S = \{\text{hot}, \text{cold}\}$ $\quad A = \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix}$ $\quad B = \begin{bmatrix} 0.05 & 0.15 & 0.8 \\ 0.75 & 0.15 & 0.1 \end{bmatrix}$

$\pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$



Hidden Markov Models

HMMs make two independence assumptions



$$P(q_t | q_1, \dots, q_{t-1}, o_1, \dots, o_{t-1}) = P(q_t | q_{t-1})$$

$$P(o_t | q_1, \dots, q_{t-1}, o_1, \dots, o_{t-1}) = P(o_t | q_t)$$

Three Tasks

- **Evaluation**: estimate the likelihood of an observation sequence

  Given an HMM μ and observation sequence Ω, determine the likelihood $\mathrm{P}(\Omega|\mu)$

- **Decoding**: find the most probable state sequence

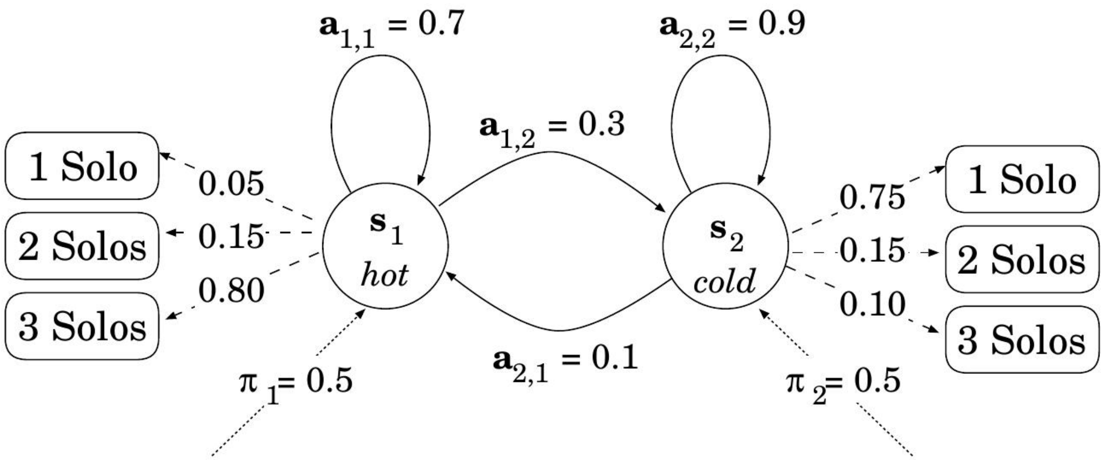
  Given an HMM μ and observation sequence Ω, determine the most probable hidden state sequence *Q*

- **Learning**: estimate parameters of HMM

  Given observation sequence Ω, the set of possible states *S* and observations *O* in an HMM, learn parameters $\mu = (A, B, \Pi)$

Evaluation

- Aim: estimate the likelihood of an observation sequence
- What is the probability of observing 3-Solos, 3-Solos, 1-Solo?



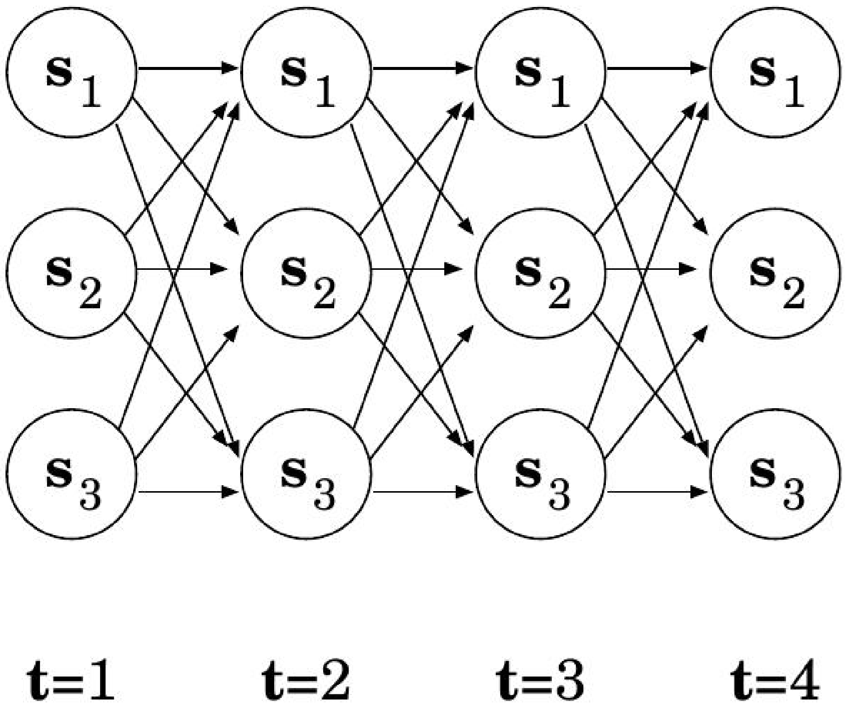
Evaluation

- What is the probability of observing 3-Solos, 3-Solos, 1-Solo?
  - if we know that the states were *hot, hot, cold*
    Easy to calculate: $\mathcal{O}(T)$
  - if we don't know the hidden state sequence
    Harder to calculate: $\mathcal{O}(TN^T)$

$T = |\Omega|$ is the length of the sequence, and $N = |S|$ is the number of states

Evaluation

- If there are 3 states and 4 time steps



Evaluation

- Probability of the state sequence

$$\mathrm{P}(Q|\mu) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

- Probability of observation sequence Ω for state sequence Q:

$$\mathrm{P}(\Omega|Q,\mu) = \prod_{t=1}^{T} P(o_t|q_t,\mu)$$

- Probability of a given observation sequence Ω considering all possible state sequences:

$$\mathrm{P}(\Omega|\mu) = \sum_{Q} \mathrm{P}(\Omega|Q,\mu)\,\mathrm{P}(Q|\mu)$$

$\mathcal{O}(TN^T)$

The Forward Algorithm

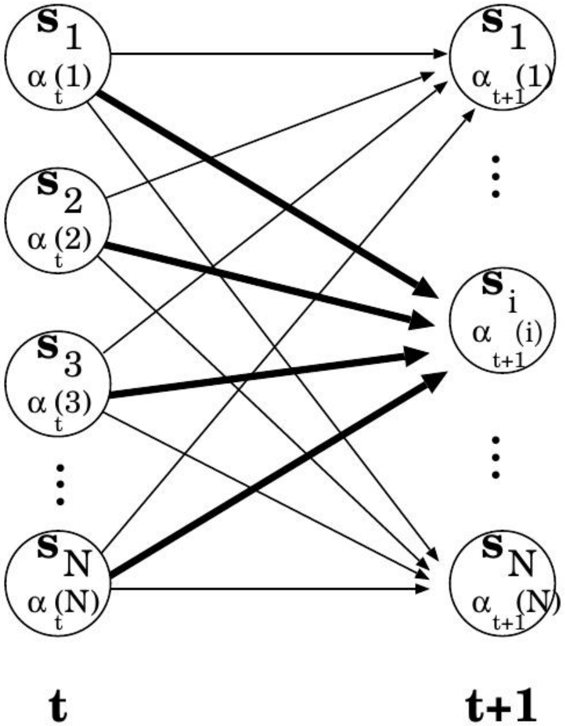
- Efficient computation of total probability $\mathrm{P}(\Omega|\mu)$ through dynamic programming
- Probability of the first t observations is the same for all possible $t+1$ length sequences
- Define **forward probability**: the probability of the partial observation sequence, $o_1, o_2, \ldots, o_t$ and state s_i at time t, given the model μ

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = s_i|\mu)$$

- By caching forward probabilities, we can avoid redundant calculations. $\mathcal{O}(TN^2)$.

The Forward Algorithm

- Store forward probabilities $\alpha_t(j)$ and reuse them to compute $\alpha_{t+1}(i)$



The Forward Algorithm

- **Initialisation:** $t = 1$, state $i = 1, \ldots, N$

$$\alpha_1(i) = \pi_i b_i(o_1)$$

- **Induction:** $t = 1, \ldots, T-1$, state $i = 1, \ldots, N$

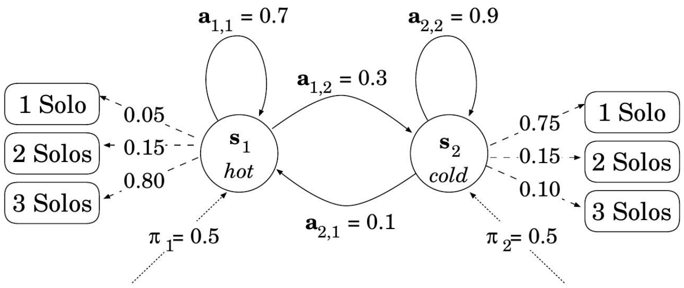
$$\alpha_{t+1}(i) = \left(\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right) b_i(o_{t+1})$$

- **Termination**

$$\mathrm{P}(\Omega|\mu) = \sum_{i=1}^{N} \alpha_T(i)$$

The Forward Algorithm: Example

- $\mathrm{P}(3\text{−Solos}, 3\text{−Solos}, 1\text{−Solo}|\mu)$?



- Initialisation/induction: $\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i|\mu)$

	$t = 1$	$t = 2$	$t = 3$
$\alpha_t(hot)$	$0.5 \times 0.8 = 0.4$	$[0.4 \times 0.7 + 0.05 \times 0.1] \times 0.8 = 0.228$	$[0.228 \times 0.7 + 0.0165 \times 0.1] \times 0.05 = 0.0080625$
$\alpha_t(cold)$	$0.5 \times 0.1 = 0.05$	$[0.4 \times 0.3 + 0.05 \times 0.9] \times 0.1 = 0.0165$	$[0.228 \times 0.3 + 0.0165 \times 0.9] \times 0.75 = 0.0624375$

- Termination:

$$\mathrm{P}(3\text{−Solos}, 3\text{−Solos}, 1\text{−Solo}|\mu) = 0.0080625 + 0.0624375 = 0.0705$$

Decoding

- Aim: find the most probable state sequence
- Given the observation 3-Solos, 3-Solos, 1-Solo, what is the most probable weather sequence?
  - Use the most probable state for each step?

→ No, cannot guarantee to find the most probable *sequence*, as the transition probability from the most likely state at $t-1$ to the most likely state at t can be small.

Decoding

- Aim: find the most probable state sequence
- Given the observation 3-Solos, 3-Solos, 1-Solo, what is the most probable weather sequence?
  - Brute-force: enumerate the probabilities of all hidden state sequences and sort, $\mathcal{O}\big(\underbrace{TN^T}_{\text{compute probabilities}} + \underbrace{N^T logN^T}_{\text{sort}}\big)$

  - More efficient method: Viterbi algorithm

The Viterbi Algorithm

- Preliminaries: define some variables
  - The maximum probability for a partial sequence $(q_1, q_2, \ldots, q_{t-1}, q_t = s_i)$ along a single path:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = s_i | \mu)$$

  - Keep track of the path: for the most probable partial sequence $(q_1, q_2, \ldots, q_{t-1}, q_t = s_i)$, the hidden state at $t - 1$ is $\psi_t(i)$

The Viterbi Algorithm

- **Initialisation**: $t = 1$, state $i = 1, \ldots, N$

$$\delta_1(i) = \pi_i b_i(o_1)$$

$$\psi_1(i) = 0$$

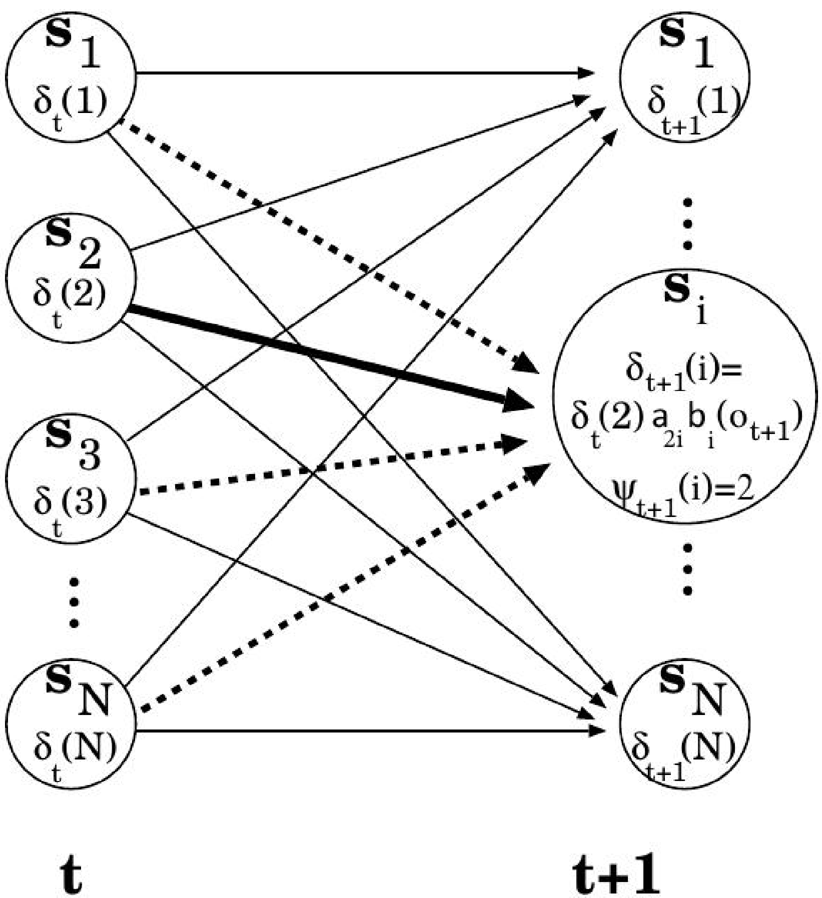
- **Induction**: $t = 1, \ldots, T-1$, state $i = 1, \ldots, N$

$$\delta_{t+1}(i) = \max_{1 \le j \le N} (\delta_t(j) a_{ji})\, b_i(o_{t+1})$$

$$\psi_{t+1}(i) = \arg\max_{1 \le j \le N} (\delta_t(j) a_{ji})$$

The Viterbi Algorithm

- **Induction**



The Viterbi Algorithm

- **Termination**

$$P_{best} = \max_{1 \le i \le N} \delta_T(i)$$

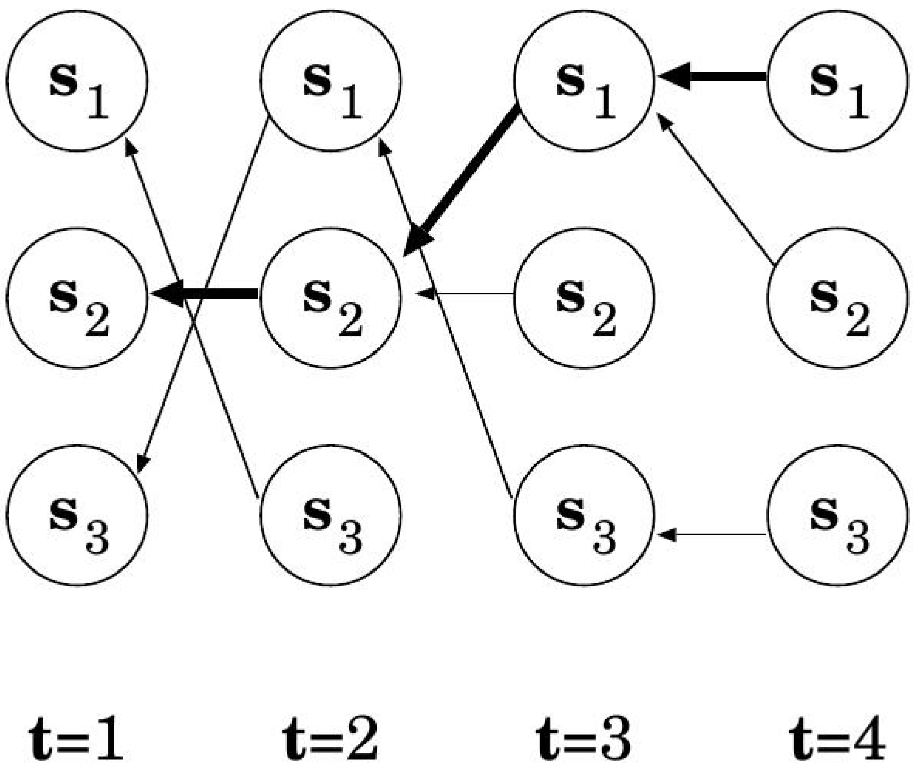
$$q_T^* = \arg\max_{1 \le i \le N} \delta_T(i)$$

- **Backtrack** to establish the best path

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \text{ for } t = T-1, T-2, \dots, 1$$

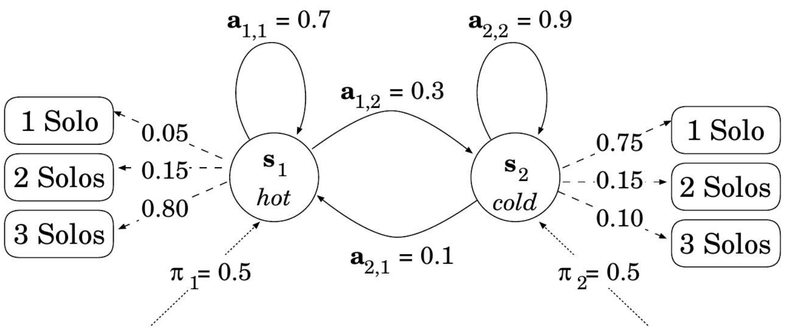
The Viterbi Algorithm

- **Backtrack**



The Viterbi Algorithm: Example

- Most probable state sequence, given the observation sequence: 3−Solos, 3−Solos, 1−Solo



- Initialisation/induction:

	$t = 1$	$t = 2$	$t = 3$
$\delta_t(hot)$ $\psi_t(hot)$	$0.5 \times 0.8 = 0.4$ 0	$\max(0.4 \times 0.7, 0.05 \times 0.1) \times 0.8 = 0.224$ $\leftarrow hot$	$\max(0.224 \times 0.7, 0.012 \times 0.1) \times 0.05 = 0.00784$ $\leftarrow hot$
$\delta_t(cold)$ $\psi_t(cold)$	$0.5 \times 0.1 = 0.05$ 0	$\max(0.4 \times 0.3, 0.05 \times 0.9) \times 0.1 = 0.012$ $\nwarrow hot$	$\max(0.224 \times 0.3, 0.012 \times 0.9) \times 0.75 = 0.0504$ $\nwarrow hot$

The Viterbi Algorithm: Example

- Termination/backtracking

$$P_{best} = 0.0504$$

$$q_T^* = cold$$

$$q_{T-1}^* = hot$$

$$q_{T-2}^* = hot$$

- The most probable sequence of hidden states for the observation sequence (3-Solos, 3-Solos, 1-Solo) is *hot, hot, cold*

Learning HMMs

- **Supervised** case: assume we have labelled data, it is possible to use simple MLE to learn the parameters of our model.

$$a_{ij} = P(s_j|s_i) = \frac{freq(s_i, s_j)}{freq(s_i)}$$

$$b_i(o_k) = P(o_k|s_i) = \frac{freq(o_k, s_i)}{freq(s_i)}$$

$$\pi_i = P(q_1 = s_i) = \frac{freq(q_1 = s_i)}{\sum_j freq(q_1 = s_j)}$$

- No state labels? **Unsupervised** case - uses forward-backward algorithm (Baum-Welch algorithm, out of scope).

Reflections

- HMM is a highly efficient approach to structured classification, but with limited representation of context (only observation and state sequences)
- HMM tends to suffer from floating point underflow
  - use scaling coefficients for Forward Algorithm
  - use logs for Viterbi Algorithm

Applications

Applications of Sequence Labelling

- **Parts-of-speech Tagging**
  - **INPUT**: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.
  - **OUTPUT**:

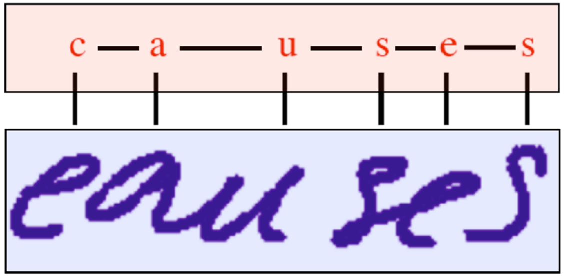
    Profits/N soared/V at/P Boeing/N Co./N,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N./.

    Labels: N = noun; V = verb; P = preposition; ADV = adverb...

Source: Collin 2013

Applications of Sequence Labelling

- **Optical Character Recognition**



Source: Hofmann 2013

Summary

- What is structured classification?
- How do we evaluate HMMs?
- How do we decode HMMs?
- How do we learn HMMs given labelled training data?
- What are the limitations of HMMs?

References

- Philip Blunsom, *Structured Classification for Multilingual Natural Language Processing*, PhD thesis, University of Melbourne, 2007.
- Michael Collin, *Tagging Problems, and Hidden Markov Models.* 2013. http://www.cs.columbia.edu/~mcollins/hmms-spring2013.pdf.
- Thomas Hofmann, *Structured Prediction (with application in information retrieval)*, 2009. http://mlg.eng.cam.ac.uk/mlss09/mlss_slides/Hoffman_1_2.pdf.
- Lawrence R. Rabiner, *A tutorial on Hidden Markov Models and selected applications in speech recognition*, Proceedings of the IEEE, 77(2), 1989.