Graphical Models

Bayesian Networks

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Describe Bayesian Networks

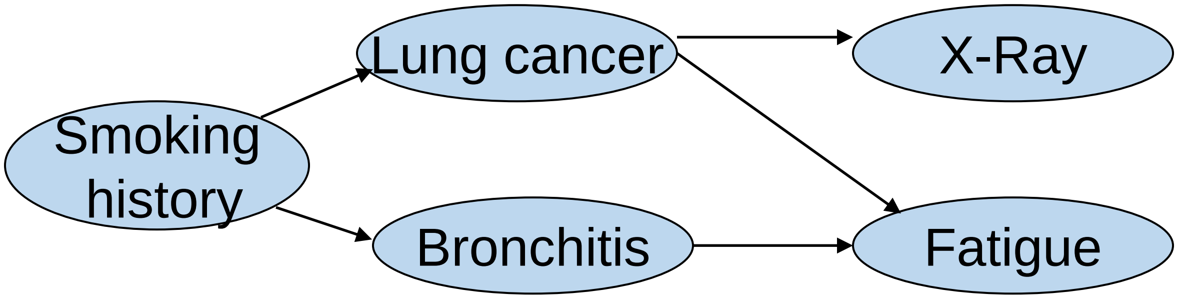
Objective

Illustrate key tasks in implementing Bayesian Networks

Why do we use graphical models?

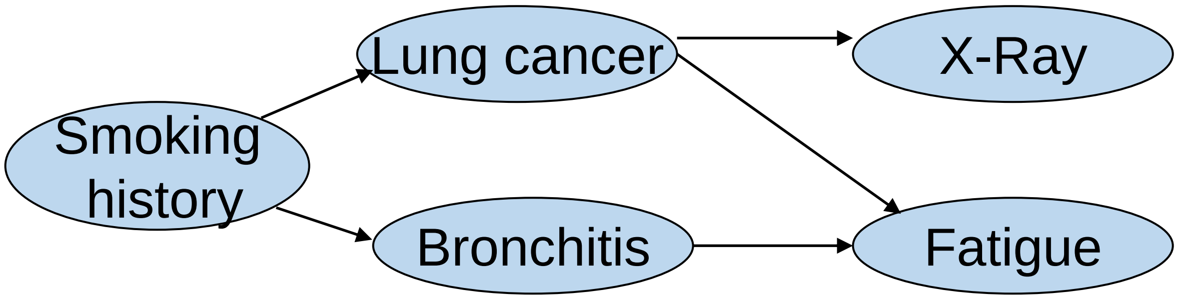
- **In machine learning, we are often concerned with joint distributions of many random variables.**

- **A graph may provide an intuitive way of representing or visualizing the relationships of the variables.**
  - Making it easier for domain experts to build a model



Graphical Models for Casual Relations

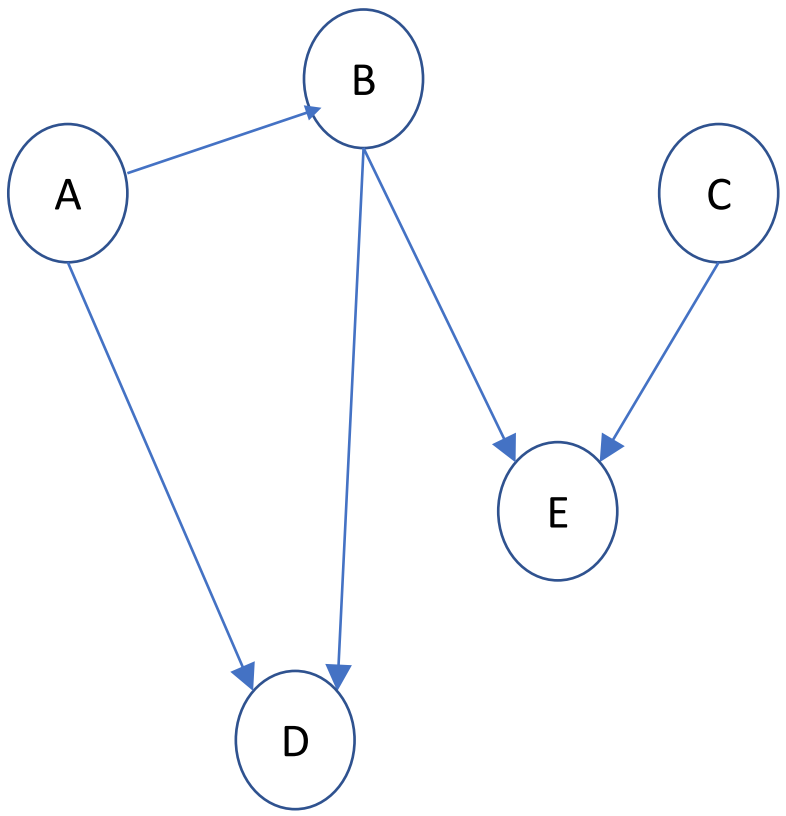
- **Graphical models arise naturally from, often causal, independency relations of physical events.**



- **Caveat: probabilistic relationship does not imply causality.**

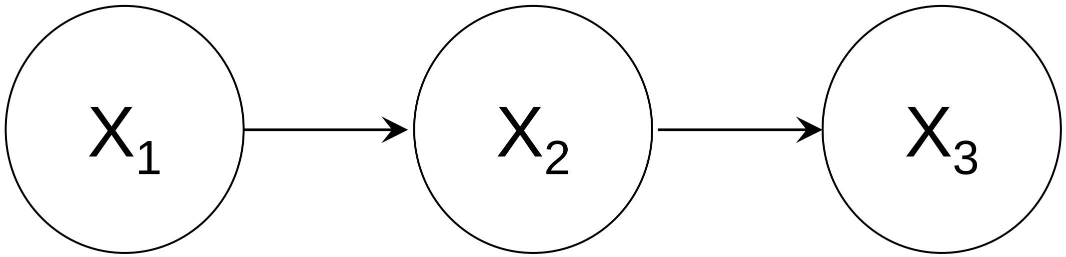
Bayesian Networks

- **A BN is directed acyclic graph (DAG), where**
  - Nodes (vertices) represent random variables.
  - Directed edges represent immediate dependence of nodes.
- **Other names: Belief networks, Bayes nets, etc.**



Conditional Independence

- **E.g., given the following graph, check the relationship between X_3 and X_1**



  - X_3 is dependent of X_2, and X_2 is dependent of X_1
  - Thus X_3 is dependent of X_1
  - But given X_2, X_3 is dependent of X_1

➔ Conditional Independence

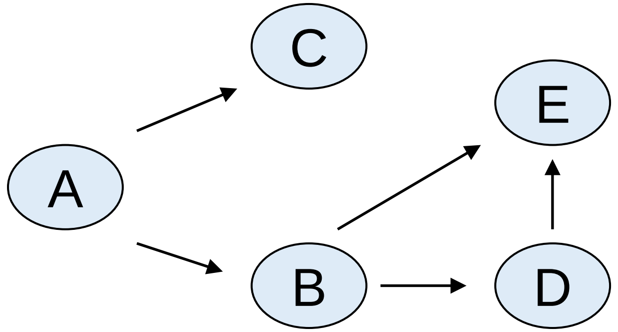
BN for General Conditional Dependency

- **A BN can be used to model given conditional dependencies**
  - For example, using the *chain rule of probability*, we have

$$P(A,B,C,D,E)=P(A)P(B|A)P(C|A,B)P(D|A,B,C)P(E|A,B,C,D)$$

- **If we know that, given A, C won't rely on B, and so forth, we may have**

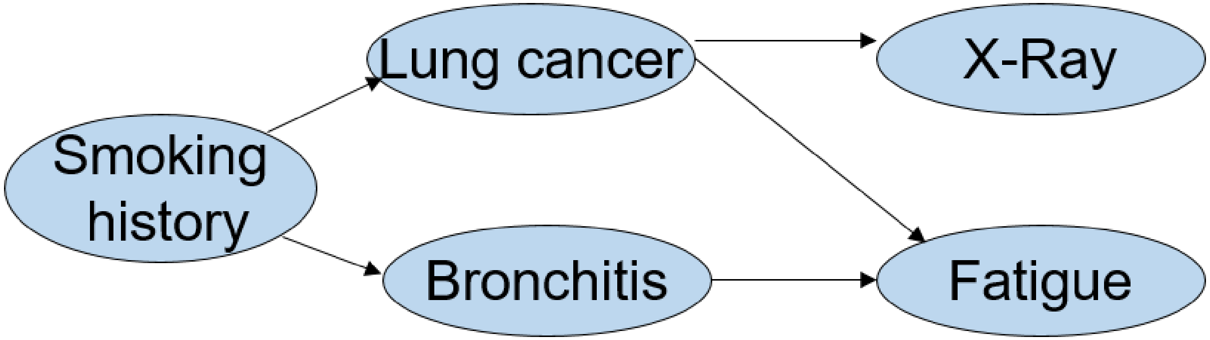
$$P(A,B,C,D,E)=P(A)P(B|A)P(C|A)P(D|B)P(E|B,D)$$



➢ We could represent joint distributions more compactly in BN → Efficient computation

Inference in Bayesian Networks

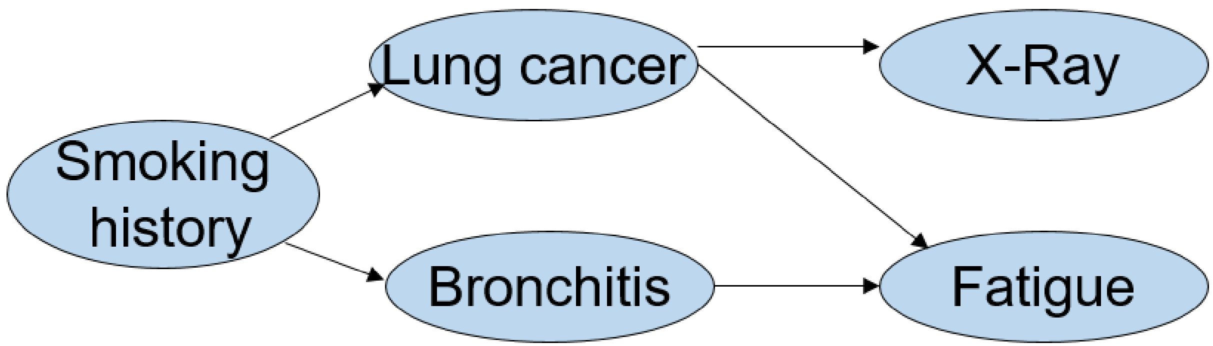
- **Given a model and some data ("evidence"), how to update our belief?**



- **What are the model parameters?**

Inference in Bayesian Networks (cont'd)

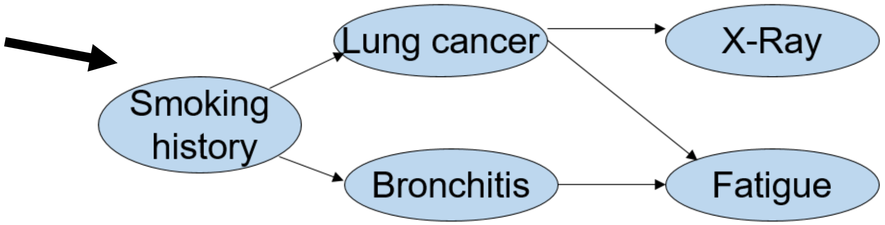
- **Given a model and some data ("evidence"), how to update our belief?**



- **E.g., for a patient with certain smoking history (non-smoker), whose X-ray result is positive, and who does not experience fatigue:**
  - What is probability of having lung cancer?

Inference in Bayesian Networks (cont'd)

- **In a simple BN like this, we can compute the exact probabilities.**



- **In general, for a tree-structured BN, we may use belief propagation for the inference problem.**
- **For general structures, sometimes it is possible to generalize the above method (e.g., the *junction tree algorithm*). More often, we must resort to approximation methods**
  - E.g. Variational methods, Sampling (Monte Carlo) methods.

Learning in Bayesian Networks

- **Learning parameters (probabilities) for a given BN (the graph is given).**
  - Estimate the (conditional) probabilities from past data.

- **Learning both the structure and the parameters for a BN**
  - A more challenging task beyond the scope of this discussion.

Learning the Probabilities

Basic ideas

- Use relative frequency for estimating probability.
- A prior distribution is typically assumed.
- The prior is then updated by the data into posterior.
- Using the MLE principle

The so-called “Expectation-Maximization (EM) Algorithm” is often used.

- Iteratively update our guess for the parameter and each step attempts to apply the MLE principle.

Graphical Models

Hidden Markov Formulation

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Introduce Hidden Markov Models

Objective

Illustrate HMM with intuitive examples

Hidden Markov Models

- **Hidden Markov Models (HMMs) are a type of dynamic Bayesian Network**
  - Modeling a process indexed by time
- **“Hidden”: the observations are due to some underlying (hidden) states not directly observable.**
- **“Markov”: the state transitions are governed by a Markov process.**

Discrete Markov Process

- **Consider a system which may be described at any time as being in one of a set of N distinct states, S_1, …, S_N.**
- **At time instances $t=1,2,3$, …, the system changes its state according to certain probability. The full description requires us to know $P(s^t=S_j \mid s^{t-1}=S_i, s^{t-2}=S_k, \ldots, s^1=S_m)$ for all $t, i, k, \ldots, m$, where s^t stands for the state of the system at time t.**
  - For a first-order Markov chain, we need to consider only

$$P(s^t=S_j \mid s^{t-1}=S_i)$$

  - Further assume *P*s are "stationary":

$$a_{ij} = P(s^t=S_j \mid s^{t-1}=S_i),\ 1 \le i,j \le N, \text{ for any } t.$$

A Simple Example

- **Assume one of the three states for each day:**

  S_1-rainy, S_2-cloudy, S_3-sunny

- **Assume the transition probability matrix**

$$A = \{a_{ij}\} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

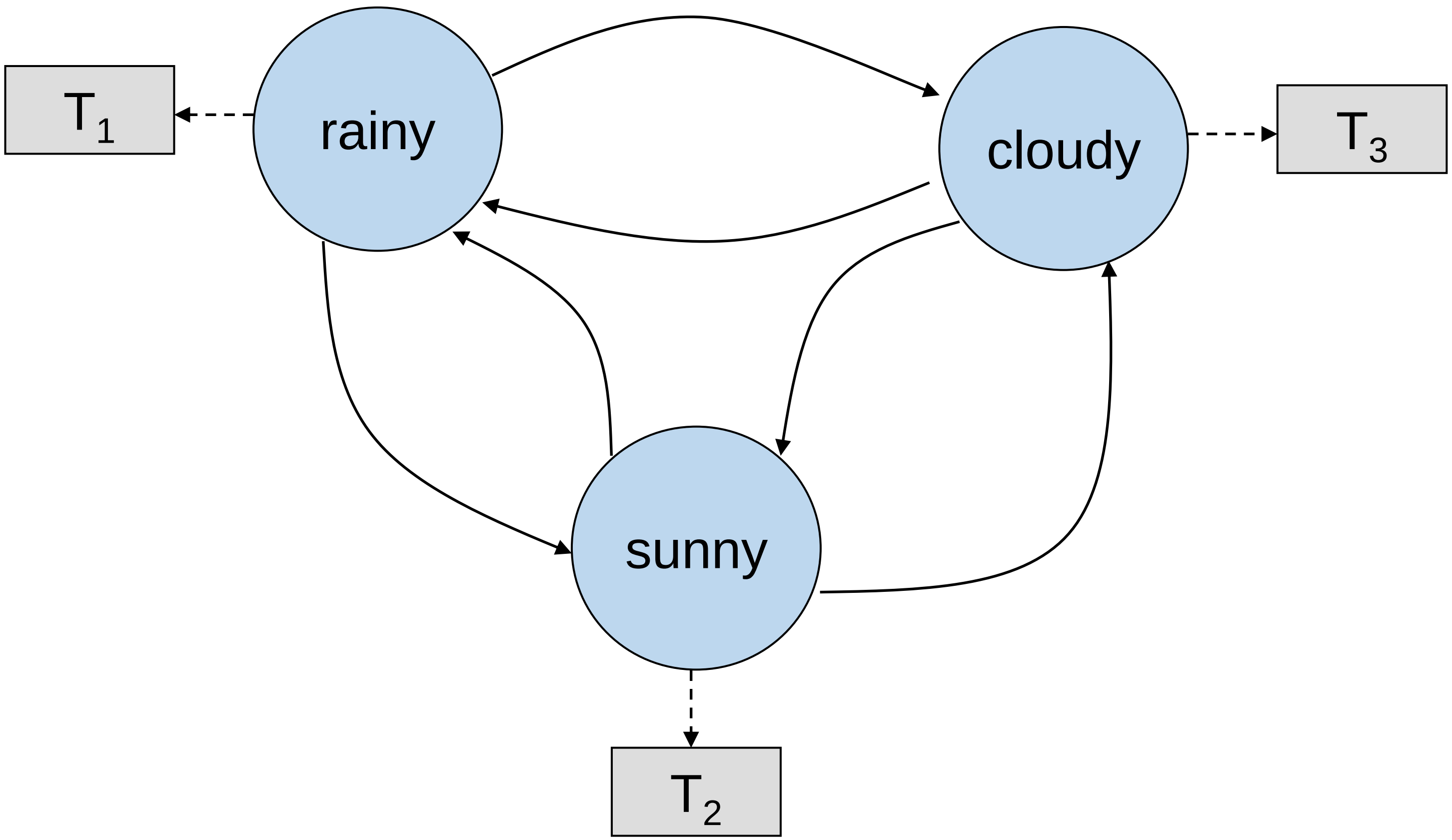
- **Many questions we may ask, based on this model.**
  - E.g., Given today is cloudy, what is the probability it remains to be cloudy for next 5 days?

Extending to “Hidden” States

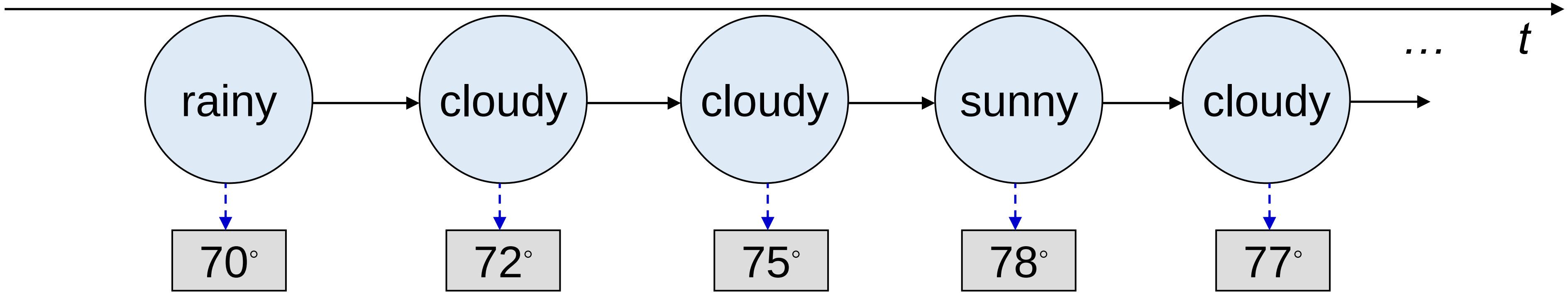
- **The previous example is an “observable” Markov model: the output of the system/process is the states of interest.**

- **Now assume that we can only measure the (average) temperature of a day**
  - Further assume this measurement is useful for predicting the weather states (rainy, cloudy, sunny).
  - We can view the temperature values as being produced by the *hidden states* of interest, i.e., the weather.

A Simple HMM



A Specific Process from the Model



Specifying an HMM

- **Θ: the set of hidden states.**

- **The state transition probabilities** $a_{ij} = P(s^t = S_j \mid s^{t-1} = S_i), 1 \le i,j \le N$
  - Let $A = \{a_{ij}\}$ be the transition probability matrix

- **Ω: the set of outputs (observations).**

Specifying an HMM (cont'd)

- **The observation probabilities: $P(o^t|s^t)$, where o^t stands for the observation at time t, given the state s^t. This is also called the emission probability.**
  - For discrete observation space, we can define $B=\{b_{jk}\}=P(o^t=\mathrm{v}_k \text{ at } t|s^t=S_j)$ as the emission probability matrix, where v_k is the k^{th} symbol in Ω

- **The initial state distribution $\pi = \{\pi_i\}$, $\pi_i=P(s^1=S_i)$**
  - Sometimes we are given an initial state, i.e., $P(s^1=S_i)=1$ for certain i.

Basic Problems in HMM

For a given HMM $\Lambda=\{\Theta,\Omega,A,B,\pi\}$

- Problem 1: Given an observation (sequence) $\boldsymbol{O}=\{o^1,o^2, \ldots ,o^k\}$, what is the most likely state sequence $\boldsymbol{S}=\{s^1,s^2, \ldots ,s^k\}$ that has produced $\boldsymbol{O}$?

- Problem 2: How likely is an observation $\boldsymbol{O}$ (i.e., what is $P(\boldsymbol{O})$) ?

- Problem 3: How to estimate the model parameters (A,B,π)?

Graphical Models

Hidden Markov Models: Learning & Inference

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Implement HMM learning & inference algorithms

Basic Problems in HMM

For a given HMM $\Lambda=\{\Theta,\Omega,A,B,\pi\}$

- Problem 1: Given an observation (sequence) $\boldsymbol{O}=\{o^1,o^2, \ldots ,o^k\}$, what is the most likely state sequence $\boldsymbol{S}=\{s^1,s^2, \ldots ,s^k\}$ that has produced $\boldsymbol{O}$?

- Problem 2: How likely is an observation $\boldsymbol{O}$ (i.e., what is $P(\boldsymbol{O})$) ?

- Problem 3: How to estimate the model parameters (A,B,π)?

Problem 1: State Estimation

- Given an observation (sequence) $O=\{o^1,o^2, \ldots ,o^k\}$, what is the most likely state sequence $S=\{s^1,s^2, \ldots ,s^k\}$ that has produced O?

- Formally, we need to solve

$$\underset{\boldsymbol{S}}{\operatorname{argmax}} P(\boldsymbol{S}|\boldsymbol{O})$$

- Or, equivalently,

$$\underset{\boldsymbol{S}}{\operatorname{argmax}} \frac{P(\boldsymbol{S},\boldsymbol{O})}{P(\boldsymbol{O})} = \underset{\boldsymbol{S}}{\operatorname{argmax}} P(\boldsymbol{S},\boldsymbol{O})$$

Problem 1: State Estimation (cont'd)

- For a given HMM, we may simplify *P*(*S*,*O*) as

$$P(\boldsymbol{S},\boldsymbol{O}) = P(\boldsymbol{O}|\boldsymbol{S})P(\boldsymbol{S})$$

$$= P(o^1 \ldots o^k|s^1 \ldots s^k)\prod_{j=1}^{k} P(s^j|s^1 \ldots s^{j-1})$$

$$\simeq P(o^1 \ldots o^k|s^1 \ldots s^k)\prod_{j=1}^{k} P(s^j|s^{j-1})$$

$$= \prod_{i=1}^{k} P(o^i|o^1 \ldots o^{i-1}, s^1 \ldots s^i)\prod_{j=1}^{k} P(s^j|s^{j-1})$$

$$\simeq \prod_{i=1}^{k} P(o^i|s^i)\prod_{j=1}^{k} P(s^j|s^{j-1}) = \prod_{i=1}^{k} P(o^i|s^i)\, P(s^i|s^{i-1})$$

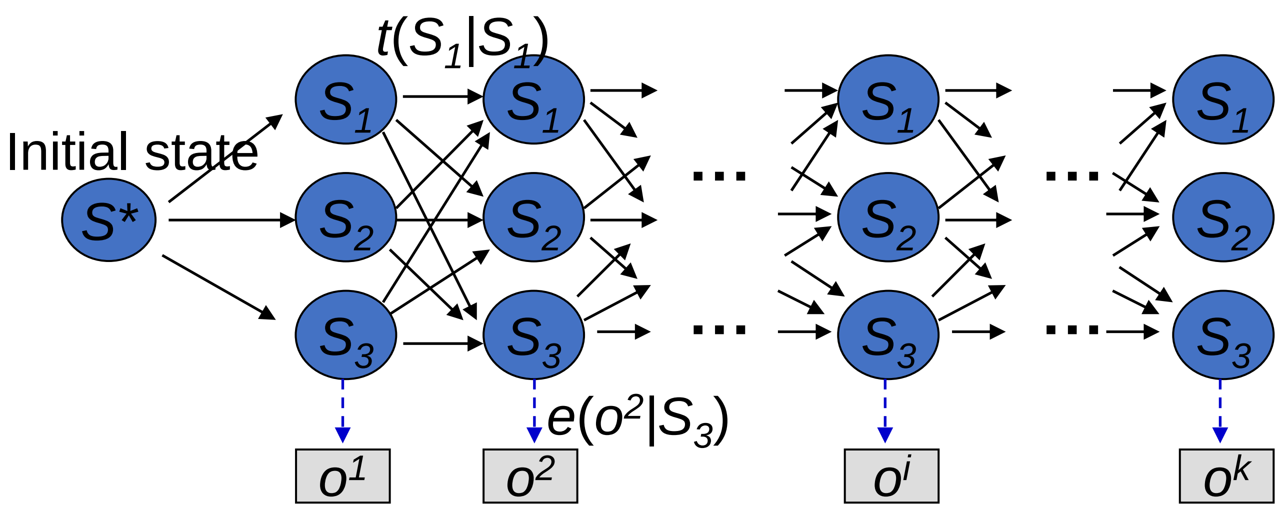
The "Weather" Example

Let's expand the state space as a trellis, for the earlier example:

S_1-rain, S_2-cloudy, S_3-sunny

-- $t(.|.)$ is the transition probability and $e(.|.)$ the emission probability.

➔ To identify a path for which the product of t's and the e's is maximized.



Viterbi Algorithm for Problem 1

A dynamic programming solution

– For each state in the trellis, we record:

1. $\delta_{s_i}(t)$ is the probability of taking the maximal path up to time *t-1* ending at state S_i at time *t* and while generating $o^1 \ldots o^t$

2. $\psi_{s_i}(t)$ is the state sequence that resulted in the maximal probability up to state S_i at time *t*.

Viterbi Algorithm (cont'd)

- **Initialization** $$\delta_{S_i}(1) = t(S_i|s^*)e(o^1|S_i), \quad \forall S_i \in \Theta$$

- **Induction:**
  for 2≤*t*≤*k*, do
  $$\delta_{S_i}(t) = \max_{S_j} t(S_i|S_j)e(o^t|S_i)\delta_{S_j}(t-1)$$
  $$\psi_{S_i}(t) = \operatorname*{argmax}_{S_j} t(S_i|S_j)e(o^t|S_i)\delta_{S_j}(t-1)$$

- **Termination:**
  - The probability of the best state sequence: $\max_{S_j} \delta_{S_j}(k)$
  - The best last state: $\hat{s}^k = \operatorname*{argmax}_{S_j} \delta_{S_j}(k)$
  - Back trace to get other states:
  $$\hat{s}^t = \psi_{\hat{s}^{t+1}}(t), \text{ for } t = k-1, \ldots, 1.$$

Problem 2: Evaluate *P*(*O*)

- To evaluate *P*(*O*), we can do $P(O) = \sum_{S} P(S, O)$

- From the trellis, a solution can be found by summing the probabilities of all paths generating the given observation sequence.

- A dynamic programming solution: <u>the forward algorithm</u> or the backward algorithm.

The Forward Algorithm

- **Define the forward probability $\alpha_{S_i}(t)$, which is the probability for all paths up to time *t-1* ending at state S_i at time *t* and generating $o^1 \ldots o^t$.**

1. Initialization: $\alpha_{S_i}(1) = t(S_i|s^*)e(o^1|S_i), \quad \forall S_i \in \Theta$

2. Induction:
   for $2 \le t \le k$, do $$\alpha_{S_i}(t) = \sum_{S_j} t(S_i|S_j)e(o^t|S_i)\alpha_{S_j}(t-1)$$

3. Termination: $$P(\boldsymbol{O}) = \sum_{S_j} \alpha_{S_j}(k)$$

Problem 3: Parameter Learning

Case 1: we have a set of labeled data – sequences in which we have the <state, observation> information

– Use relative frequency for estimating the probabilities

→ the MLE solution

$$t(S_i|S_j) = \frac{\text{number of } (s^t = S_i, s^{t-1} = S_j)}{\text{number of } S_j} \qquad e(o_r|S_j) = \frac{\text{number of } (o^t = o_r, s^t = S_j)}{\text{number of } S_j}$$

Case 2: we have only the observation sequence

– The Forward-Backward Algorithm (a.k.a. Baum-Welch Algorithm): An EM approach.