COMP4650 / COMP6490 Document Analysis 2018

Information Extraction

Gabriela Ferraro

Overview of IE lectures

- Introduction to Information Extraction (IE)
- Sequence labeling methods

    Markov Process

    The HMM algorithm

    The CRF algorithm
- Automatic summarization

* Acknowledgement: Some of the content originates from the Stanford NLP course at Coursera.org

Sequence labeling

Sequential data

- Speech, text, video analysis, time-series, stock market, genes...

Sequential labeling problem

- Is a type of pattern recognition task that involves the algorithmic assignment of a categorical label to each member of a sequence of observed values

Sequential methods

- Probabilistic methods; usually make a Markov assumption
- Algorithms: HMM, Maximum Entropy, Conditional Random Fields

Sequence labeling in NLP

speech recognition

part-of-speech tagging

sentence segmentation

grapheme to morpheme convertion

chunking (shallow syntactic parsing)

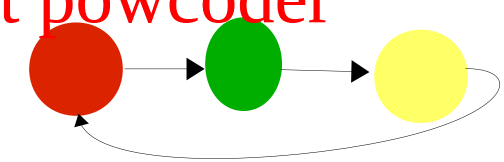
named entity recogntion

information extraction

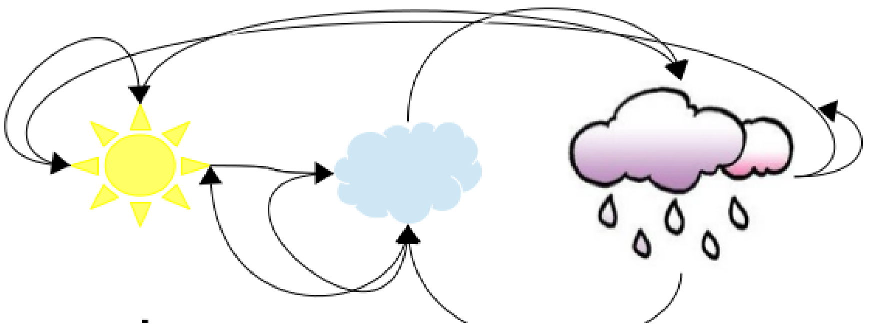
Markov Process

Deterministic patterns

- Each state is dependent solely on the previous state
- Easy to understand and determine once the transitions are fully known, e.g., semaphore



Andrei Andreyevich Markov
1856-1922



Non deterministic patterns

- It is possible for any state to follow another, e.g., weather

Markov chain

Stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event

First-order Markov chain, the prob. Of a particular state dependes only on the previous state

Markov assumption

$$P(q_i = a | q_1 \ldots q_{i-1}) = P(q_i = a | q_{i-1})$$

A **Markov chain**: compute a prob. for a sequence of events that we can be observe in the world.

But some events are not directable observable in the world…

Hidden Markov model

Hidden Markov Model

Markov assumption

$$P(q_i = a | q_1 \ldots q_{i-1}) = P(q_i = a | q_{i-1})$$

Output independence assumption: the prob of an output observation o_i depends only on the state that produce the observation q_i and not on any other state or any other observations

$$P(o_i | q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i | q_i)$$

Weather and Ice Cream

Jason Eisner, 2002

- You are a climatologist in the year 2799 studying global warming

- You can't find any records of the weather in Baltimore for summer of 2018

- But you find JE's diary
- Which lists how many ice-creams Jason ate every day that summer
- Use the observations (ice-cream ate) to estimate the temperature every day

Hidden Markov Model

Various examples exist where the process states are not directly observable, but are indirectly observable,

then we have a **Hidden Markov Model**

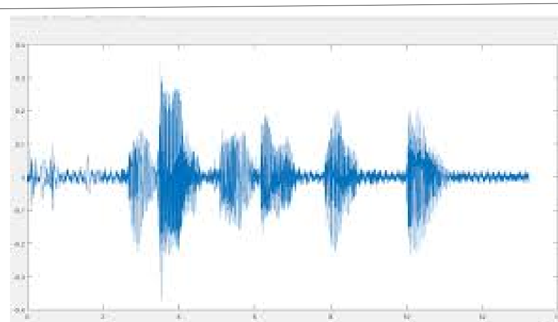
DT NN VB Adj

The house is red.

Hidden states

Observations

The house is red.

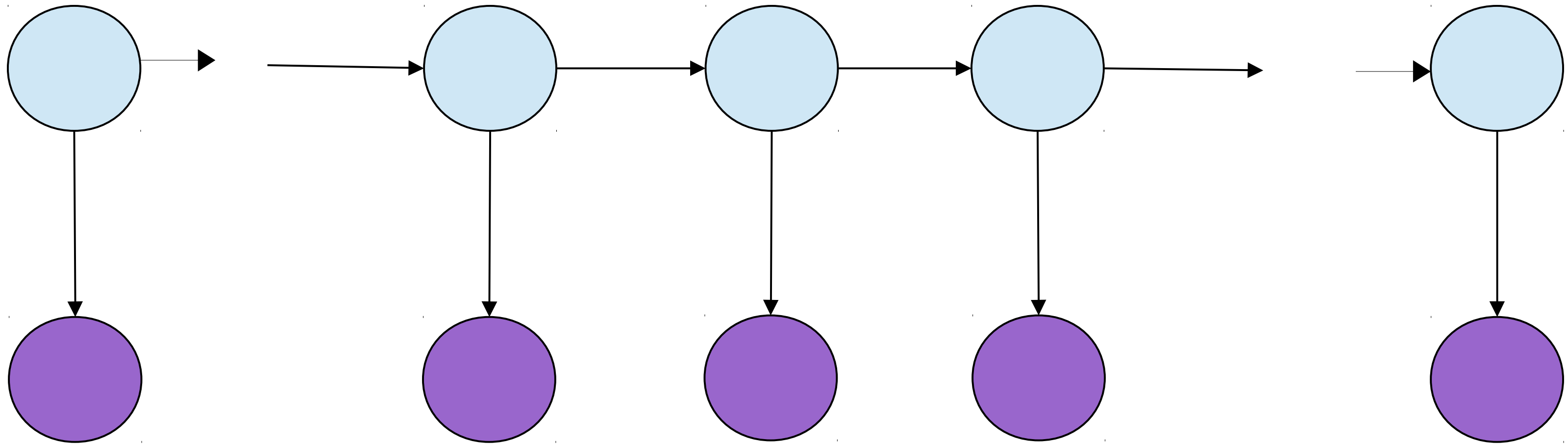


Hidden states

Observations

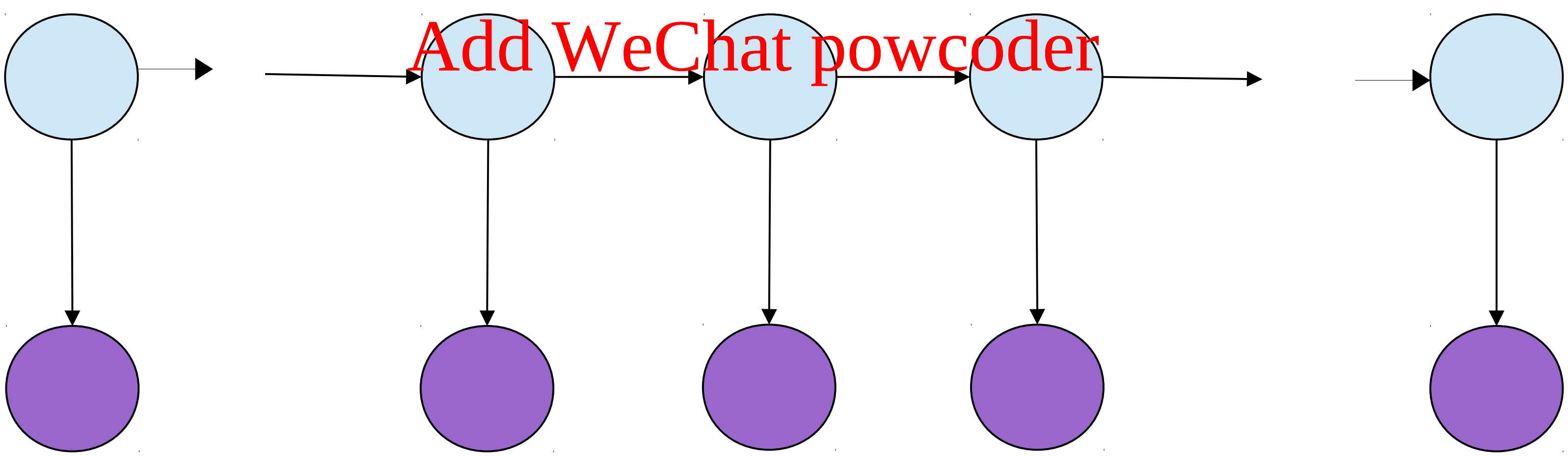
What is a Hidden Mark Model?

- HMM is a graphical model
- Circles represent states
- Arrows represent probabilistic dependencies between states



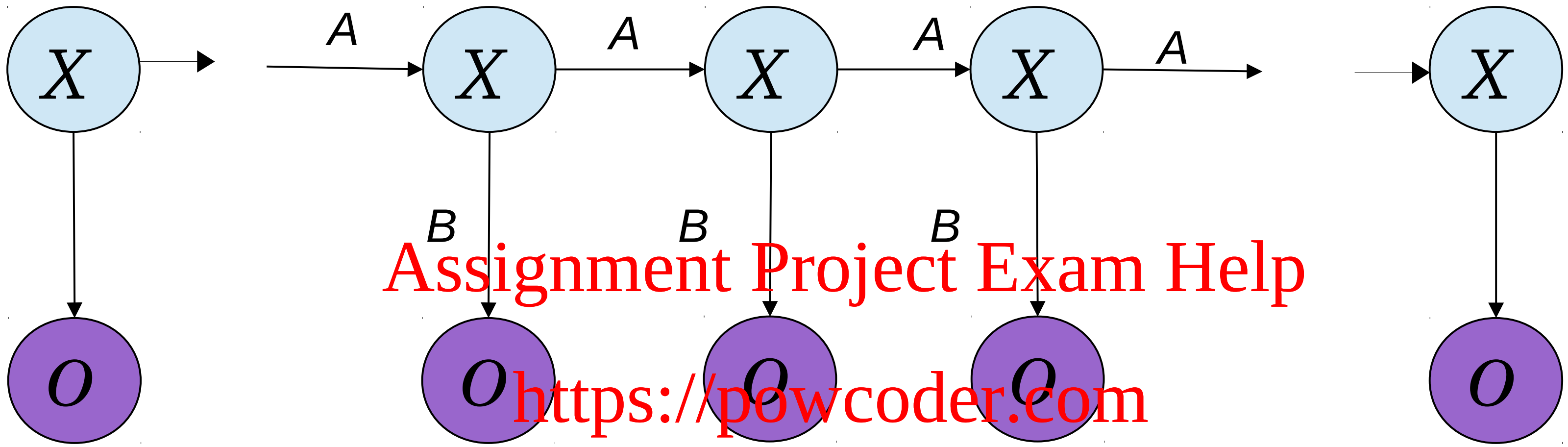
HMM Notation

- Light blue nodes are **hidden states**
  - Dependent only on the previous state
- Purple nodes are **observations states**
  - Dependent only on their hidden state



The future is independent of the past, given the present

HMM notation



{X, O, Π, *A*, *B*}

X : $\{x_1 \ldots x_N\}$ are the values for the hidden states

O : $\{o_1 \ldots o_M\}$ are the values for the observations

Parameters

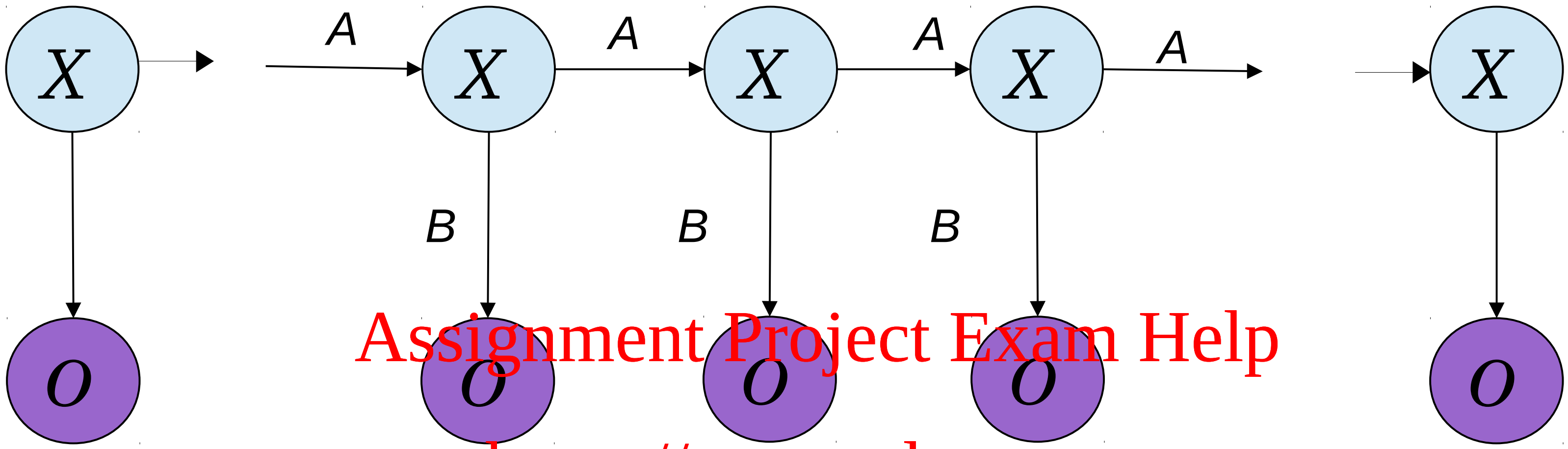
Π = $\{\pi_\iota\}$ are the initial state probabilities

A = $\{a_{ij}\}$ are the state transition probabilities

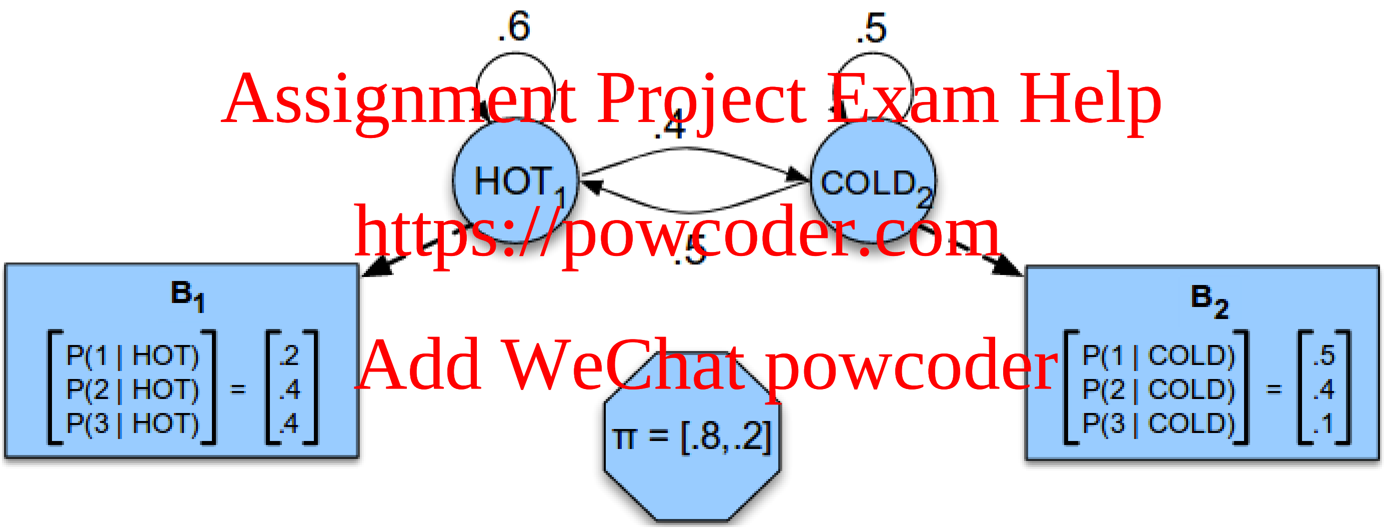
B = $\{b_{ik}\}$ are the observation state probabilities (emission)

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

HMM model $\mu=(A,B,\pi)$



$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$


.6
.5
.4
HOT1
COLD2
.5
B1
P(1 | HOT)
P(2 | HOT)
P(3 | HOT)
=
.2
.4
.4
B2
P(1 | COLD)
P(2 | COLD)
P(3 | COLD)
=
.5
.4
.1
π = [.8,.2]

HMM Problems

There are three fundamental problems that can be solved using HMM

1. **LIKELIHOOD** (testing)**:** Given an HMM model $\mu=(A,B,\pi)$ and an observation sequence ***O***, compute the likelihood $P(O|\mu)$.

   `Given # ice-creams, what is the weather?`

2. **DECODING:** Given an observation sequence ***O*** and an HMM model $\mu=(A,B,\pi)$, discover the

   best hidden state sequence ***Q***.

   `Given a sequence of ice-creams, what was the most likely weather on those days?`

3. **LEARNING:** Given an observation sequence ***O*** and set of possible states in the HMM, learn the HMM parameters ***A*** and ***B***.

Likelihood

Likelihood: Given an HMM λ = (A,B) and an observation sequence O, determine the likelihood P(O, λ)

– E.g. what is the probability of an ice-cream sequence 3 – 1 – 3?

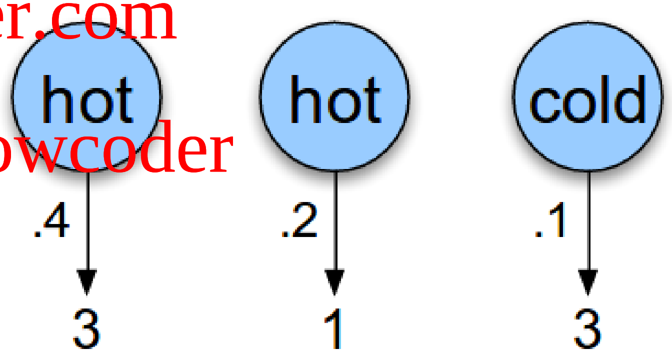
– But we don't know what the hidden state sequence is...

Likelihood

Let's make it simpler.

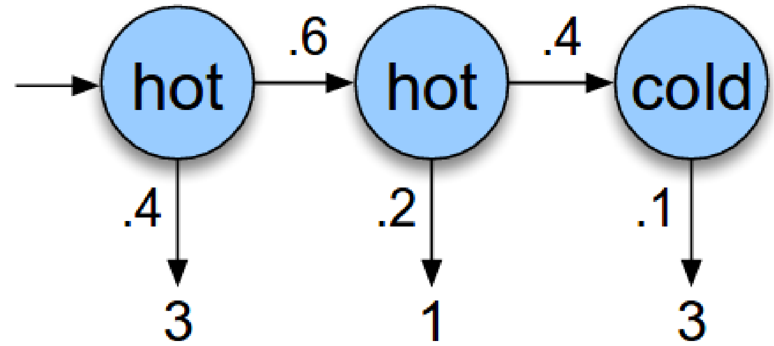
What is the likelihood of an ice-cream observed sequence 3-1-3, given the hidden state sequence *HOT HOT COLD?*

$$P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i)$$



$$P(3\ 1\ 3|\text{hot hot cold}) = P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold})$$

Likelihood



Join prob. Of been in a particular weather sequence Q and generate a particular sequence of ice-creams events

$$P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(o_i|q_i) \times \prod_{i=1}^{T} P(q_i|q_{i-1})$$

$$\begin{aligned} P(3\ 1\ 3, \text{hot hot cold}) \; &= \; P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \\ &\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \end{aligned}$$

Likelihood

Compute the prob. of ice-cream events 3-1-3 by summing over all possible weather sequences, weighted by their probability

$$P(O) = \sum_{Q} P(O,Q) = \sum_{Q} P(O|Q)P(Q)$$

$$P(3\ 1\ 3) = P(3\ 1\ 3, \text{cold cold cold}) + P(3\ 1\ 3, \text{cold cold hot}) + P(3\ 1\ 3, \text{hot hot cold}) + \ldots$$

For *N* hidden states and observation sequence of *T* observations, there are N^T possible hidden state sequences.

When *N* and *T* are large → intractable

Likelihood → Forward algorithm

Dynamic Programming algorithm, stores table of intermediate values so it need not recompute them.

Computes P(O) by summing over probabilities of all hidden state paths that could generate the observation sequence 3-1-3:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

The previous forward path probability $\alpha_{t-1}(i)$

The transition probability from the previous state to the current state α_{ij}

The state observation likelihood of the observation o_t given the current state j $b_j(o_t)$

function FORWARD(*observations* of len T, *state-graph* of len N) **returns** *forward-prob*

create a probability matrix *forward[N,T]*

for each state s **from** 1 **to** N **do** ; initialization step

$\quad$ $forward[s,1] \leftarrow \pi_s * b_s(o_1)$

for each time step t **from** 2 **to** T **do** ; recursion step

$\quad$ **for** each state s **from** 1 **to** N **do**

$$forward[s,t] \leftarrow \sum_{s'=1}^{N} forward[s',t-1] * a_{s',s} * b_s(o_t)$$

$$forwardprob \leftarrow \sum_{s=1}^{N} forward[s,T] \qquad \text{; termination step}$$

return *forwardprob*

Figure A.7 The forward algorithm, where $forward[s,t]$ represents $\alpha_t(s)$.

from https://web.stanford.edu/~jurafsky/slp3/A.pdf

HMM Problems

1. **LIKELIHOOD** (testing): Given an HMM model $\mu=(A,B,\pi)$ and an observation sequence ***O***, compute the likelihood ***P(O|μ)***.

    `Given # ice-creams, what is the weather?`

2. **DECODING:** Given an observation sequence ***O*** and an HMM model $\mu=(A,B,\pi)$, discover the best hidden state sequence ***Q***.

    `Given a sequence of ice-creams, what was the most likely weather on those days?`

3. **LEARNING:** Given an observation sequence ***O*** and set of possible states in the HMM, learn the HMM parameters ***A*** and ***B***.

Decoding: Viterbi algorithm
(Andrew Viterbi, 1967)

Decoding: Given an observation sequence *O* and an HMM $\lambda = (A,B)$, discover the ***best*** hidden state sequence of weather states in *Q*

*Given the observations 3 – 1 – 1 and an HMM, what is the **best** (most probable) hidden weather sequence of {H,C}*

Viterbi algorithm

- Dynamic programming algorithm
- Uses a dynamic programming trellis to store probabilities that the HMM is in state *j* after seeing the first *t* observations, for all states *j*

Decoding: Viterbi algorithm

Decoding: Given an observation sequence *O* and an HMM $\lambda = (A,B)$, discover the ***best*** hidden state sequence of weather states in *Q*

*Given the observations 3 – 1 – 1 and an HMM, what is the **best** (most probable) hidden weather sequence of {H,C}*

Viterbi algorithm

- Dynamic programming algorithm
- Uses a dynamic programming trellis to store probabilities that the HMM is in state *j* after seeing the first *t* observations, for all states *j*

– Value in each cell computed by taking MAX over all paths leading to this cell – i.e. best path

– Extension of a path from state i at time t-1 is computed by multiplying:

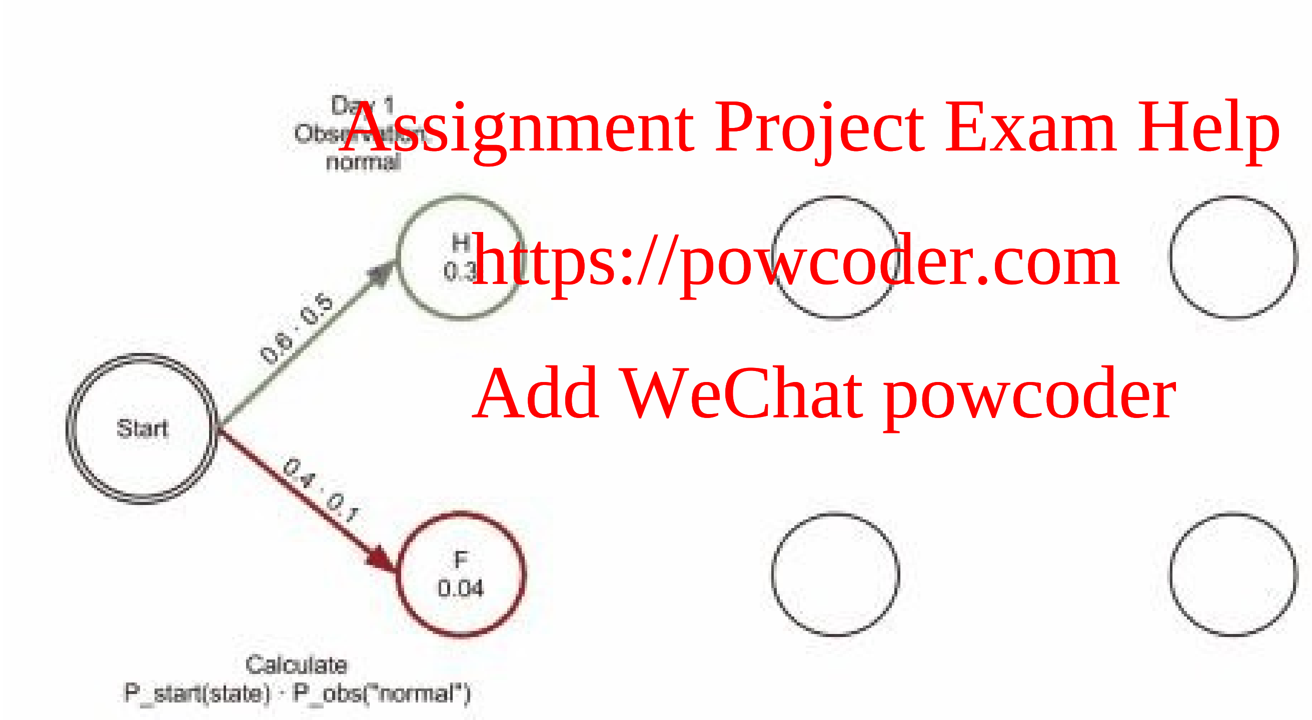
$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

– Most probable path is the max over all possible previous state sequences

Like Forward Algorithm, but it takes the max over previous path probabilities rather than sum

Viterbi example



HMM model was develop by Baum and collegues in Princeton (Baym and Petrie, 1966; Baum and Eagon, 1967)

Viterbi

Multiple independent discovery and publications

Citation	Field
Viterbi (1967)	information theory
Vintsyuk (1968)	speech processing
Needleman and Wunsch (1970)	molecular biology
Sakoe and Chiba (1971)	speech processing
Sankoff (1972)	molecular biology
Reichert et al. (1973)	molecular biology
Wagner and Fischer (1974)	computer science

Andrea Giacomo Viterbi, 1935 (age 83)

HMM Problems

There are three fundamental problems that can be solved using HMM

1. **LIKELIHOOD** (testing)**:** Given an HMM model $\mu=(A,B,\pi)$ and an observation sequence ***O***, compute the likelihood ***P*(*O*|μ)**.

   Given # ice-creams, what is the weather?

2. **DECODING:** Given an observation sequence ***O*** and an HMM model $\mu=(A,B,\pi)$, discover the

   best hidden state sequence ***Q***.

   Given a sequence of ice-creams, what was the most likely weather on those days?

3. **LEARNING/TRAINING:** Given an observation sequence ***O*** and set of possible states in the HMM, learn the HMM parameters ***A*** and ***B***.

Training: The Forward-Backward (Baum-Welch) Algorithm

- ***Learning***: Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A (transition) and B (emission)

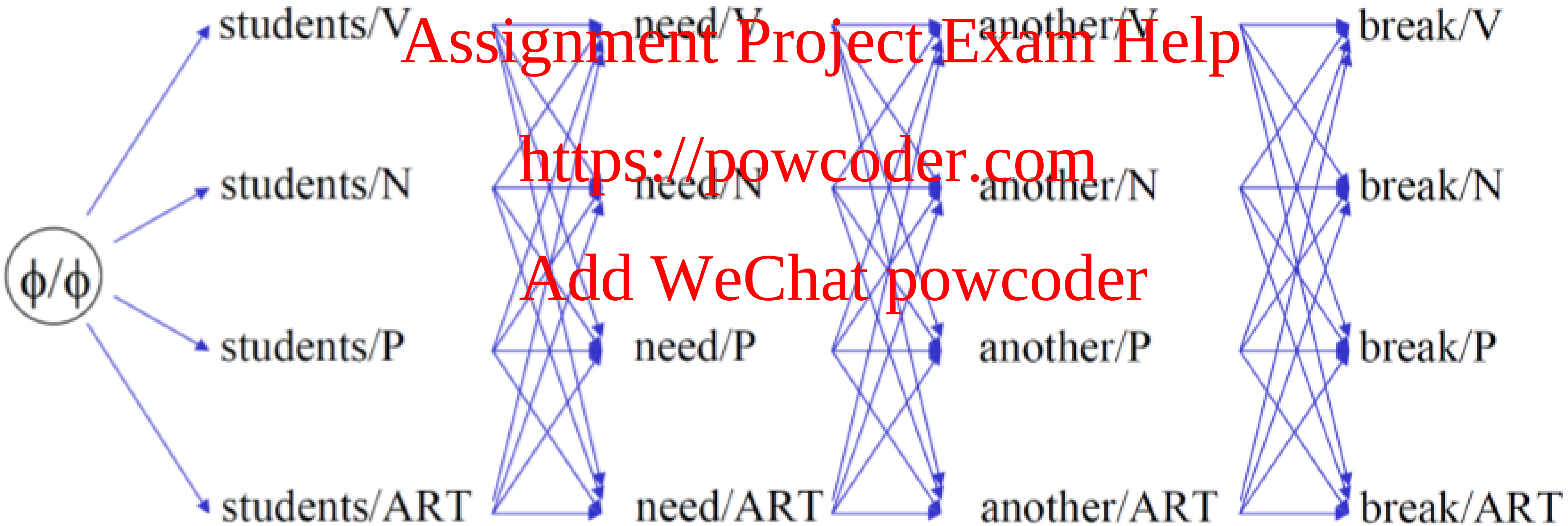
- Input: unlabeled seq of observations O and vocabulary of possible hidden states Q
  - E.g. for ice-cream weather:
    - Observations = {1,3,2,1,3,3,…}
    - Hidden states = {H,C,C,C,H,C,….}

- Intuitions
  - Iteratively re-estimate counts, starting from an initialization for A and B probabilities, e.g. all equi-probable

  - Estimate new probabilities by computing forward probability for an observation, dividing prob. mass among all paths contributing to it, and computing the backward probability from the same state

Details: https://web.stanford.edu/~jurafsky/slp3/A.pdf

POS-tagging with HMM



Summary

- HMMs are a major tool for relating observed sequences to hidden information that explains or predicts the observations

- Forward, Viterbi, and Forward-Backward Algorithms are used for computing likelihoods, decoding, and training HMMs

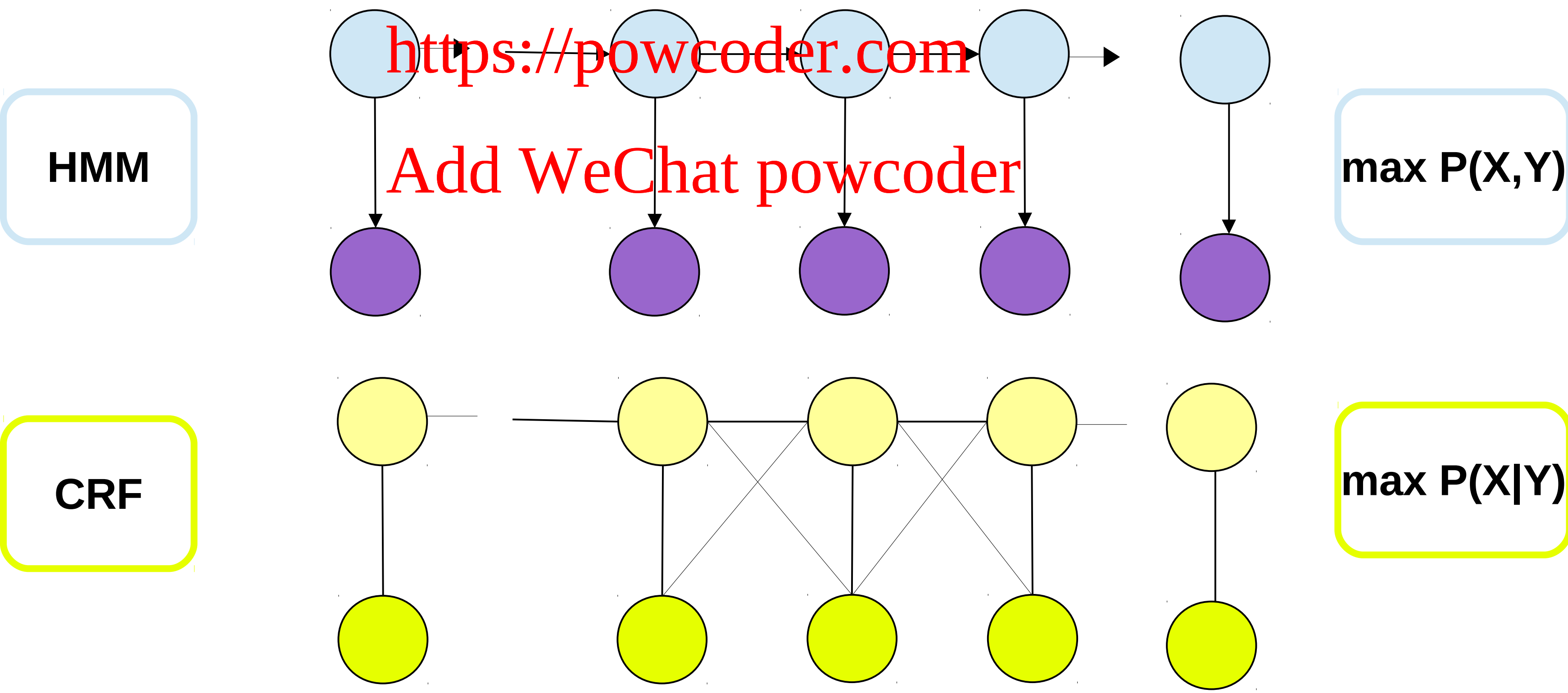
The power of HMMs

We can use the special structure of this model to do a lot of neat math and solve problems that are otherwise not solvable!!

- **NLP applications**
  - Speech Recognition
  - POS-Tagging
  - Information Extraction
  - Word/clause segmentation
- **Other applications**
  - Gene finding
  - Robot localization
  - User modeling
- **Limitations**
  - Local features
  - Simple HMM models do not work well with large data
  - Difficult to incorporate a diverse set features
  - No suited to work with long distance dependencies (up to ~3/5 grams)

Conditional Random Fields (CRF)

- CRF is a graphical model (Lafferty, McCallum, and Pereira, 2001)
- Relax the strong independence assumptions made in models such as HMM
- The biggest advantage of CRFs over HMMs is that they can handle ***overlapping features***



HMM vs. CRF

HMM

- Trained by maximizing likelihood of data and class *p(x, y)*
- Features are assumed independent
- Feature weights set independently
- Normalization is per state

CRF

- Trained by maximizing conditional likelihood of classes *p(x|y)*
- Dependency on features taken account by feature weights
- Feature weights are set mutually
- Normalization over the whole sequence

Take away

Sequential data

- Speech, text, video analysis, time-series, stock market, genes…

Sequential labeling problem

- Is a type of pattern recognition task that involves the algorithmic assignment of a categorical label to each member of a sequence of observed values

Sequential methods

- Probabilistic methods; usually make a Markov assumption, i.e. that the choice of label directly dependent only on the immediately adjacent labels;
- Algorithms: HMM, Maximum Entropy, Conditional Random Field

Markov Process are the basics for:

Reinforcement learning; Planing; RNN; Sequence2Sequence models, etc.

Anecdotal References

Markov Chains

https://www.youtube.com/watch?v=o-jdJxXL_W4

HMM 3D Simulator

https://www.youtube.com/watch?v=Fy6tLBzXT4M

HMM @ Numb3rs: Find a missing path in a map

https://www.youtube.com/watch?v=RFCMoQ4H2Hg

Viterbi algorithm @ Numb3rs: predict the next action of a criminal

https://www.youtube.com/watch?v=NdOm8NE0qD4

They always say practice makes perfect

HMM in Python, with scikit-learn

https://github.com/hmmlearn/hmmlearn

UMDHMM

http://www.kanungo.com/software/software.html#umdhmm

CRFsuite Python

http://www.chokkan.org/software/crfsuite/

CRF++ C++

http://crfpp.googlecode.com/svn/trunk/doc/index.html

GRMM Java

http://mallet.cs.umass.edu/grmm/index.php

Further References

Stamp, 2012. ***A Revealing Introduction to Hidden Markov Models****.*

Lafferty, McCallum and Pereira, 2001. ***Conditional Random Fields: Probabilistic Models for Segmentation and Labeling Sequence Data***.

Sutton and McCallum, 2006. ***An Introduction to Conditional Random Fields for Relational Learning***.

Bikel, 1999. ***An Algorithm that Learns What's in a Name***.

Bach and Badaskar, 2007. ***A survey on Relation Extraction.***