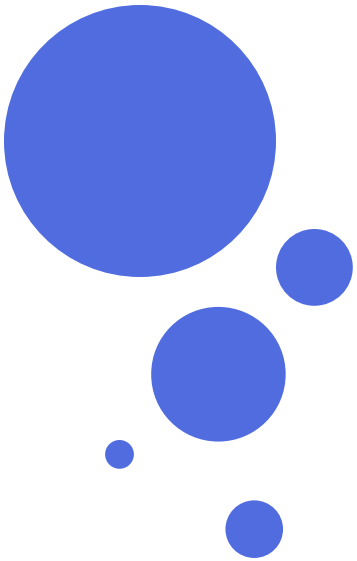
Rensselaer



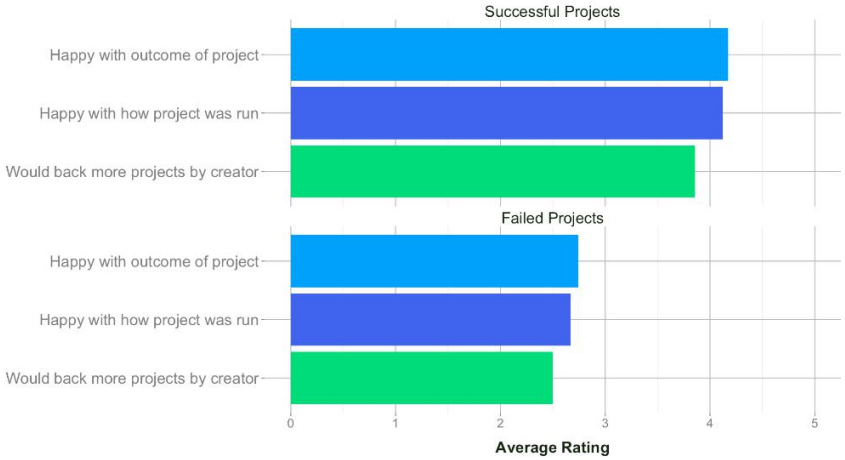
Lecture 15: Basic graph concepts, Belief Network and HMM

Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at RPI.
Email: **longc3@rpi.edu**

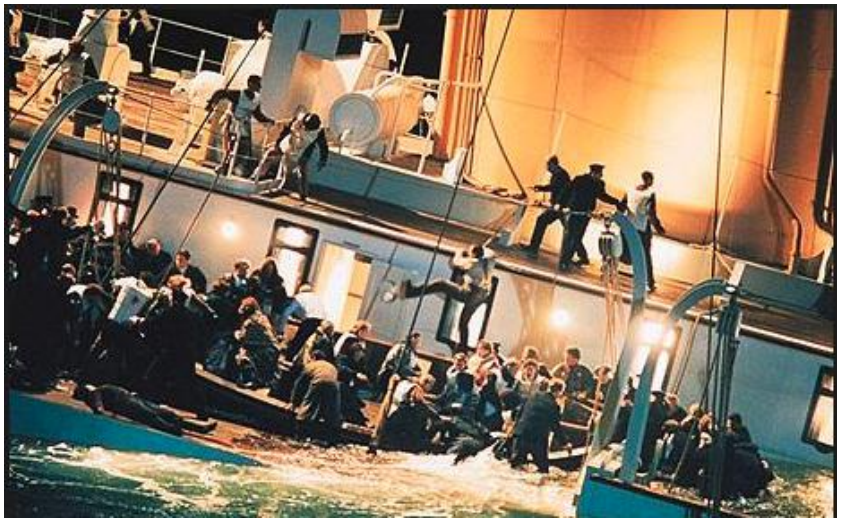
About Final Projects

No.	Project name	Authors
1	Neural Style Transfer for Video	Sarthak Chatterjee and Ashraful Islam
2	Kickstarter: succeed or fail?	Jeffrey Chen and Steven Sperazza
3	Head Pose Estimation	Lisa Chen
4	Feature selection	Zijun Cui
5	Human Face Recognition	Chao-Ting Hsueh, Huaiyuan Chu, Yilin Zhu
6	Tragedy of Titanic: a person on board can survive or not.	Ziyi Wang, Dewei Hu
7	Character Recognition	Xiangyang Mou, Tong Jian
8	Classifying groceries by image using CNN	Rui Li, Yan Wang
9	Facial expressions expression	Cameron Mine
10	Handwritten digits recognition	Kimberly Oakes

About Final Projects: Binary Classification

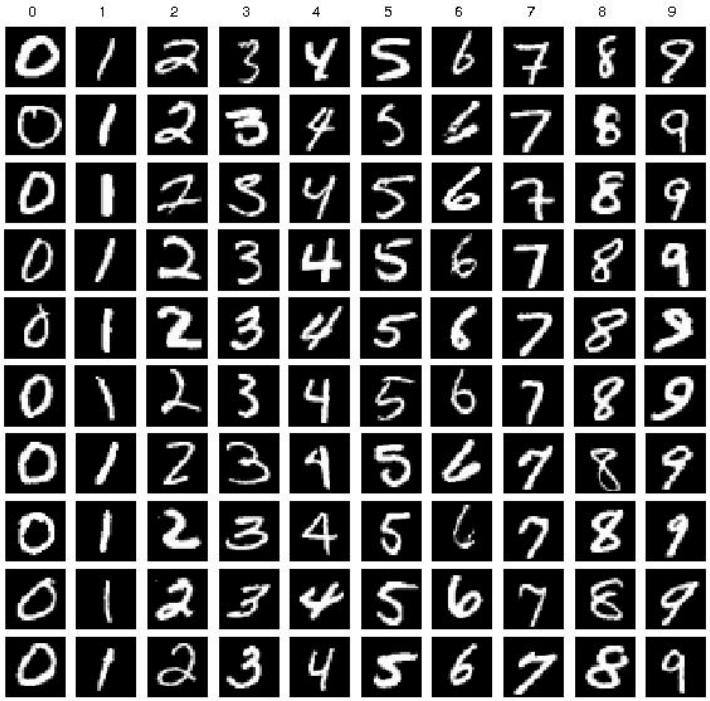


Kickstarter: succeed or fail?
Jeffrey Chen and Steven Sperazza

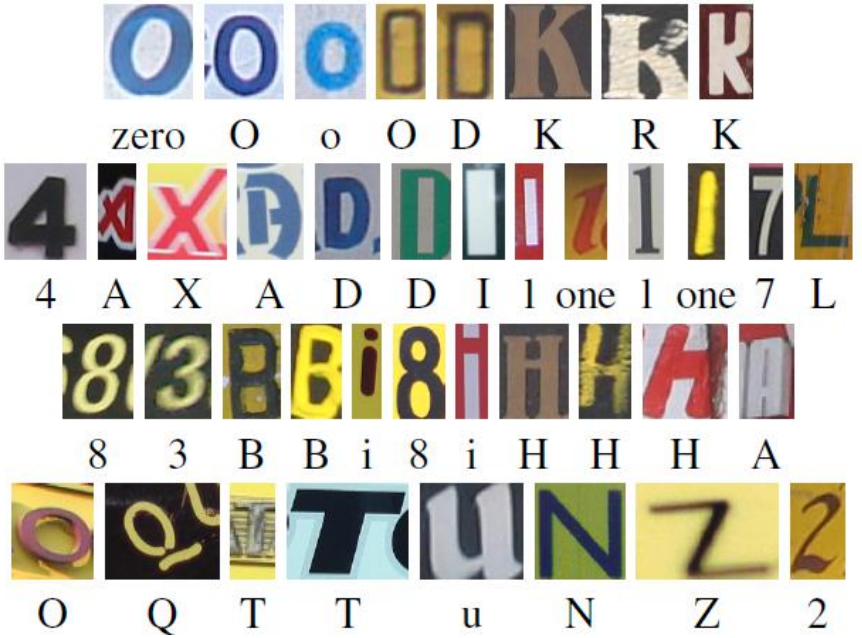


Tragedy of Titanic: a person on board can survive or not.
Ziyi Wang, Dewei Hu

About Final Projects: Multi-class Classification

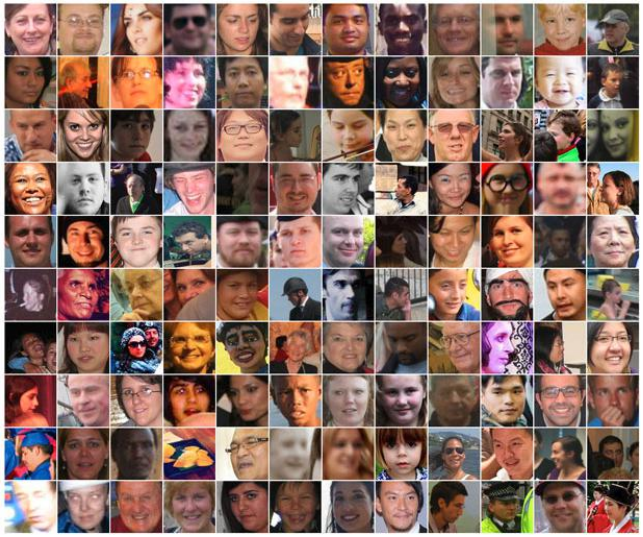


Handwritten digits recognition
Kimberly Oakes

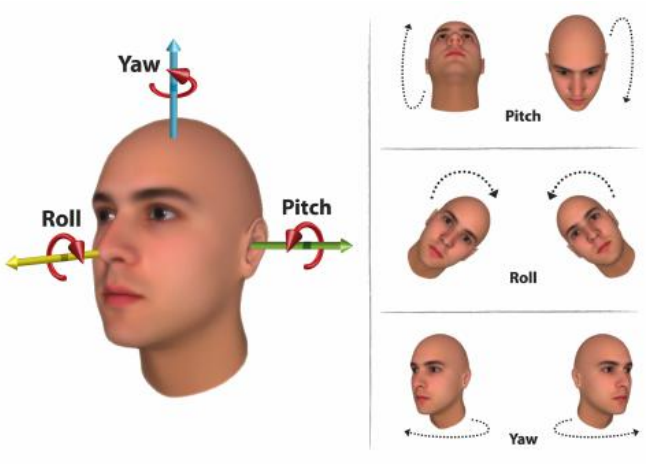


Character Recognition
Xiangyang Mou, Tong Jian

About Final Projects: Multi-class Classification



Human Face Recognition
Chao-Ting Hsueh, Huaiyuan Chu, Yilin Zhu

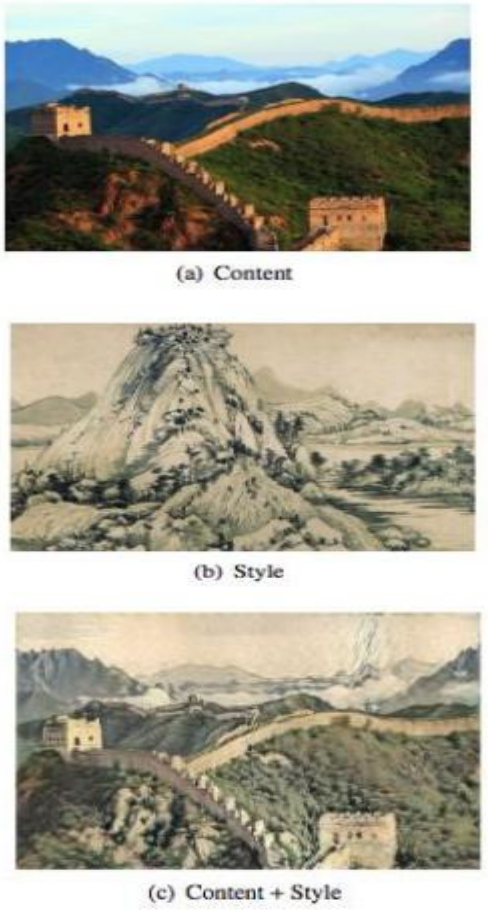


Head Pose Estimation
Lisa Chen

Facial expressions expression
Cameron Mine

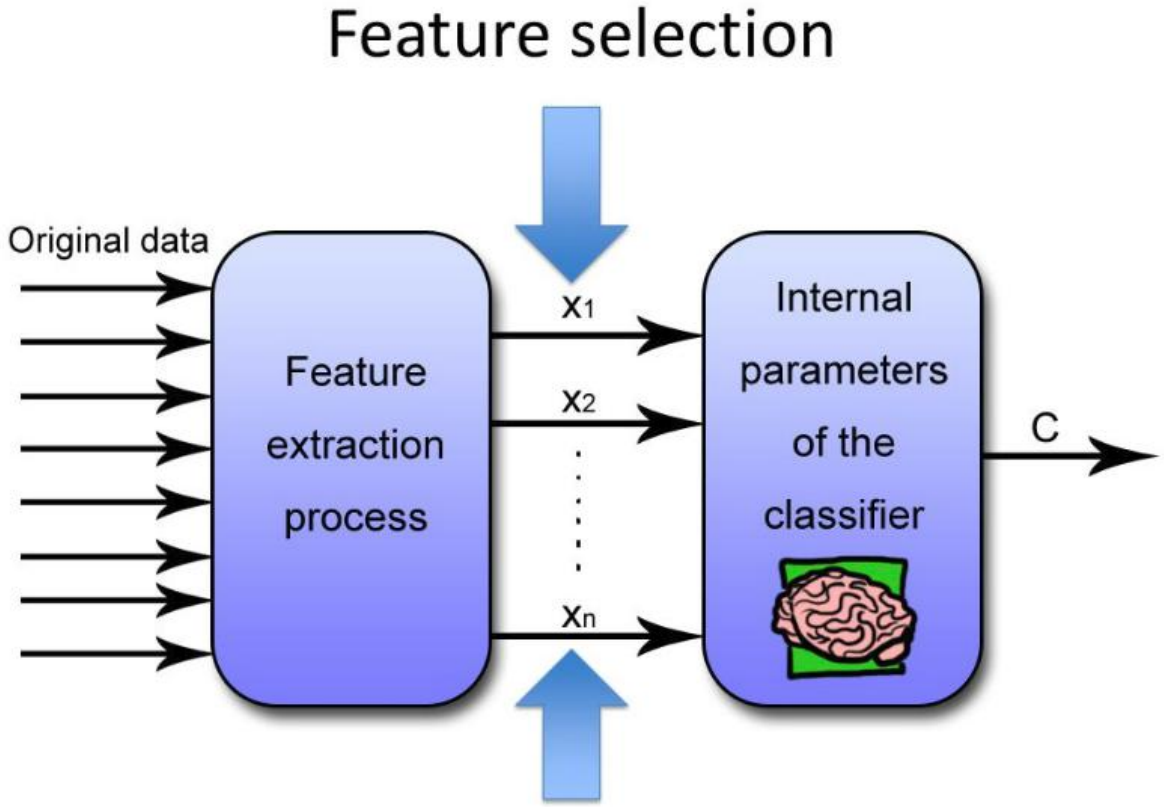
About Final Projects: CNN and GAN

Classifying groceries by image using CNN
Rui Li, Yan Wang



Neural Style Transfer for Video
Sarthak Chatterjee and Ashraful Islam

About Final Projects: Feature Selection



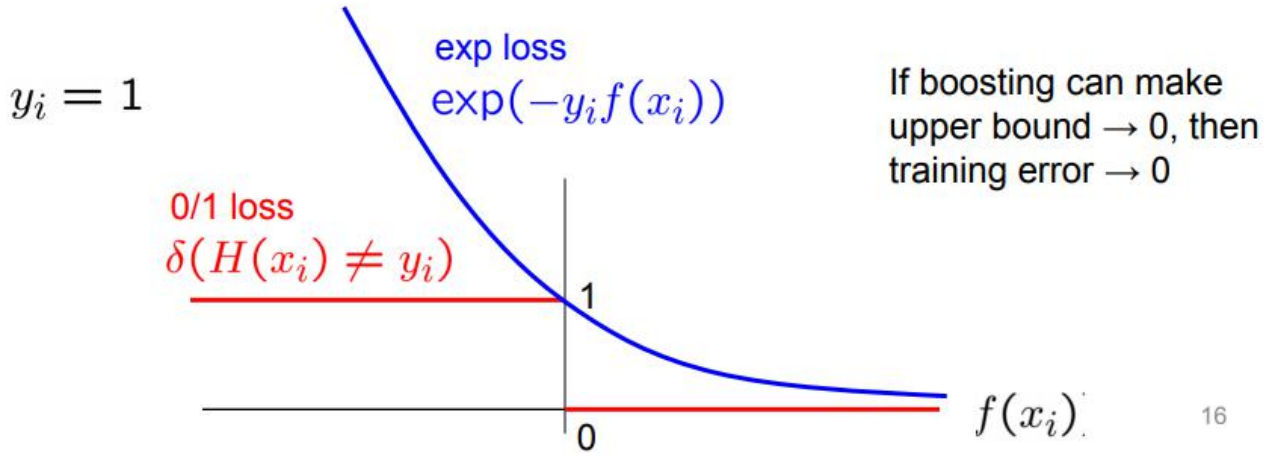
Feature selection
Zijun Cui

Guideline for the proposal presentation

- Briefly introduce the importance of project - 1 slide
- Define the problem and the project objectives -1 or 2 slides
- Investigate the related work - 1 slide
- Propose your feasible solutions and the necessary possible baselines - 1 to 3 slides
- Describle the data sets you plan to use - 1 or 2 slides
- List your detailed progress plan to complete the final project - 1 slide.
- List the references. - 1 slide

5-8 min presentation, including Q&A. I would like to recommend you to use informative figures as possible as you can to share what you are going to do with the other classmates.

Recap Previous Lecture



If boosting can make upper bound → 0, then training error → 0

16

$$\frac{1}{m}\sum_{i=1}^{m}\delta(H(x_i)\neq y_i) \leq \prod_t Z_t = \prod_t \sqrt{1-(1-2\epsilon_t)^2}$$

$$\leq \exp\left(-2\sum_{t=1}^{T}(1/2-\epsilon_t)^2\right)$$

grows as ϵ_t moves away from 1/2

Outline

- Introduction to Graphical Model
- Introduction to Belief Networks
- Hidden Markov Models

Outline

- **Introduction to Graphical Model**
- Introduction to Belief Networks
- Hidden Markov Models

Graphical Models

- GMs are graph based representations of various factorization assumptions of distributions
  - These factorizations are typically equivalent to independence statements amongst (sets of) variables in the distribution
- Directed graphs model conditional distributions (e.g. Belief Networks)
- Undirected graphs represented relationships between variables (e.g. neighboring pixels in an image)

Definition

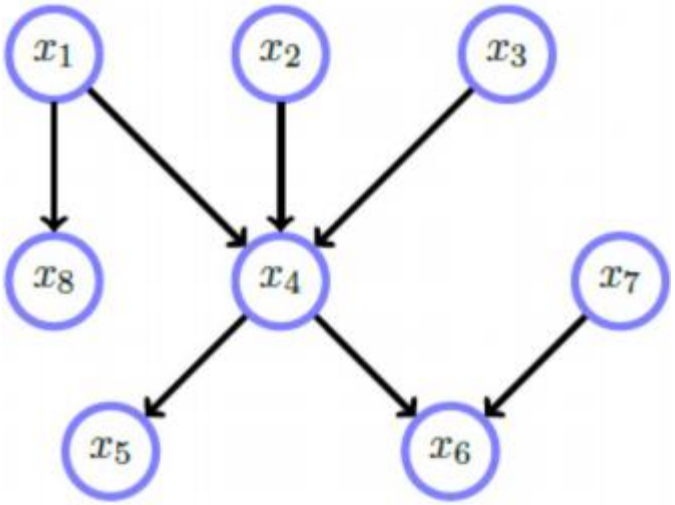
- A graph G consists of nodes (also called vertices) and edges (also called links) between the nodes
- Edges may be directed (they have an arrow in a single direction) or undirected
  - Edges can also have associated weights
- A graph with all edges directed is called a directed graph, and one with all edges undirected is called an undirected graph

More Definitions

- A **path** A–>B from node A to node B is a sequence of nodes that connects A to B
- A **cycle** is a directed path that starts and returns to the same node
- **Directed Acyclic Graph (DAG)**: A DAG is a graph G with directed edges (arrows on each link) between the nodes such that by following a path of nodes from one node to another along the direction of each edge no path will revisit a node

More Definitions

- The parents of $x4$ are $pa(x4) = \{x1, x2, x3\}$
- The children of $x4$ are $ch(x4) = \{x5, x6\}$
- Graphs can be encoded using the edge list $L=\{(1,8),(1,4),(2,4)\ldots\}$ or the adjacency matrix



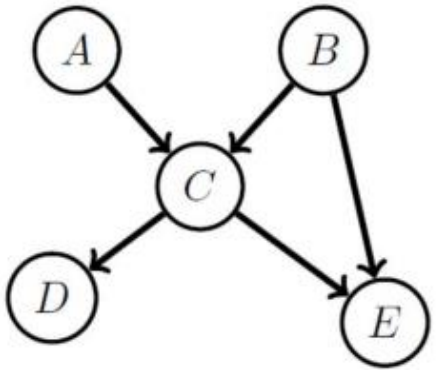
Outline

- Introduction to Graphical Model
- **Introduction to Belief Networks**
- Hidden Markov Models

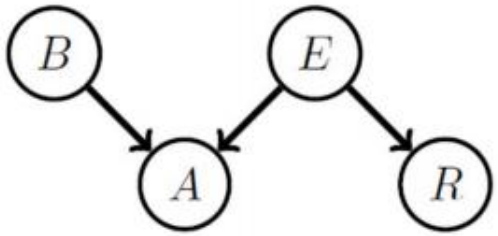
Belief Networks (Bayesian Networks)

- A belief network is a directed acyclic graph in which each node has associated the conditional probability of the node given its parents
- The joint distribution is obtained by taking the product of the conditional probabilities:

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



Alarm Example



$p(A|B,E)$

Alarm = 1	Burglar	Earthquake
0.9999	1	1
0.99	1	0
0.99	0	1
0.0001	0	0

$p(R|E)$

Radio = 1	Earthquake
1	1
0	0

The remaining data are p(B = 1) = 0.01 and p(E = 1) = 0.000001

Alarm Example: Inference

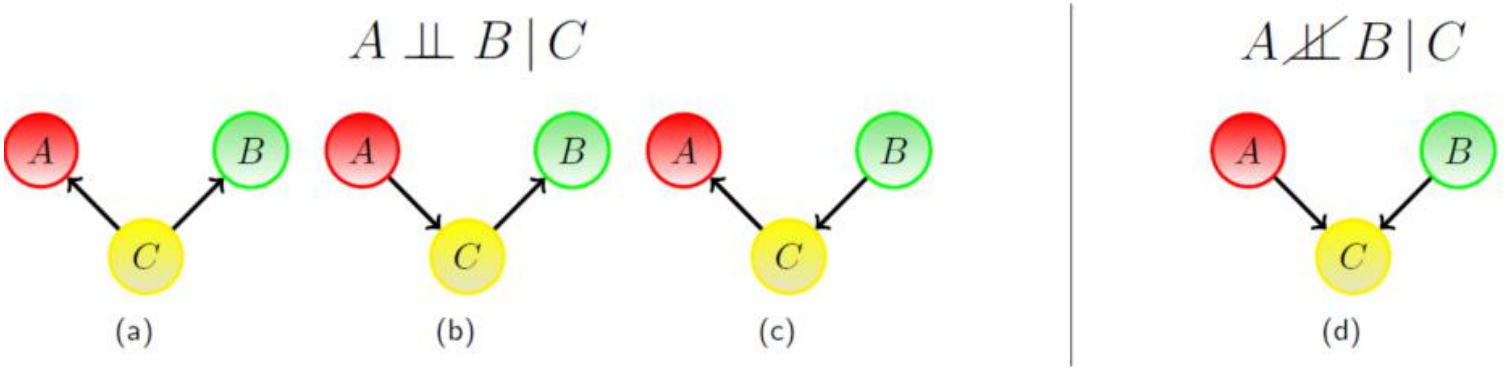
- Initial evidence: the alarm is sounding

$$p(B=1|A=1) = \frac{\sum_{E,R} p(B=1,E,A=1,R)}{\sum_{B,E,R} p(B,E,A=1,R)}$$
$$= \frac{\sum_{E,R} p(A=1|B=1,E)p(B=1)p(E)p(R|E)}{\sum_{B,E,R} p(A=1|B,E)p(B)p(E)p(R|E)} \approx 0.99$$

Alarm Example: Inference

- Additional evidence: the radio broadcasts an earthquake warning
  - A similar calculation gives $p(B = 1 \mid A = 1, R = 1) \approx 0.01$
  - Initially, because the alarm sounds, Sally thinks that she's been burgled. However, this probability drops dramatically when she hears that there has been an earthquake.
  - The earthquake `explains away' to an extent the fact that the alarm is ringing

Independence in Belief Networks



- In (a), (b) and (c), A, B are conditionally independent given C

$$\text{(a)}\quad p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(A|C)p(B|C)p(C)}{p(C)} = p(A|C)p(B|C)$$

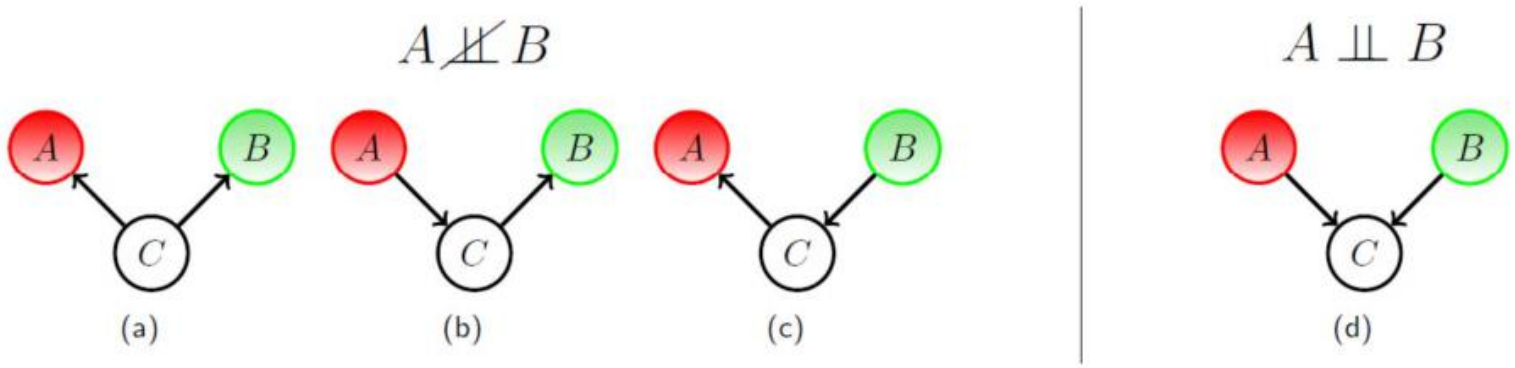
$$\text{(b)}\quad p(A,B|C) = \frac{p(A)p(C|A)p(B|C)}{p(C)} = \frac{p(A,C)p(B|C)}{p(C)} = p(A|C)p(B|C)$$

$$\text{(c)}\quad p(A,B|C) = \frac{p(A|C)p(C|B)p(B)}{p(C)} = \frac{p(A|C)p(B,C)}{p(C)} = p(A|C)p(B|C)$$

- In (d) the variables A,B are conditionally dependent given C

$$p(A,B|C) \propto p(C|A,B)p(A)p(B)$$

Independence in Belief Networks



- In (a), (b) and (c), A, B are marginally dependent
- In (d) the variables A, B are marginally independent

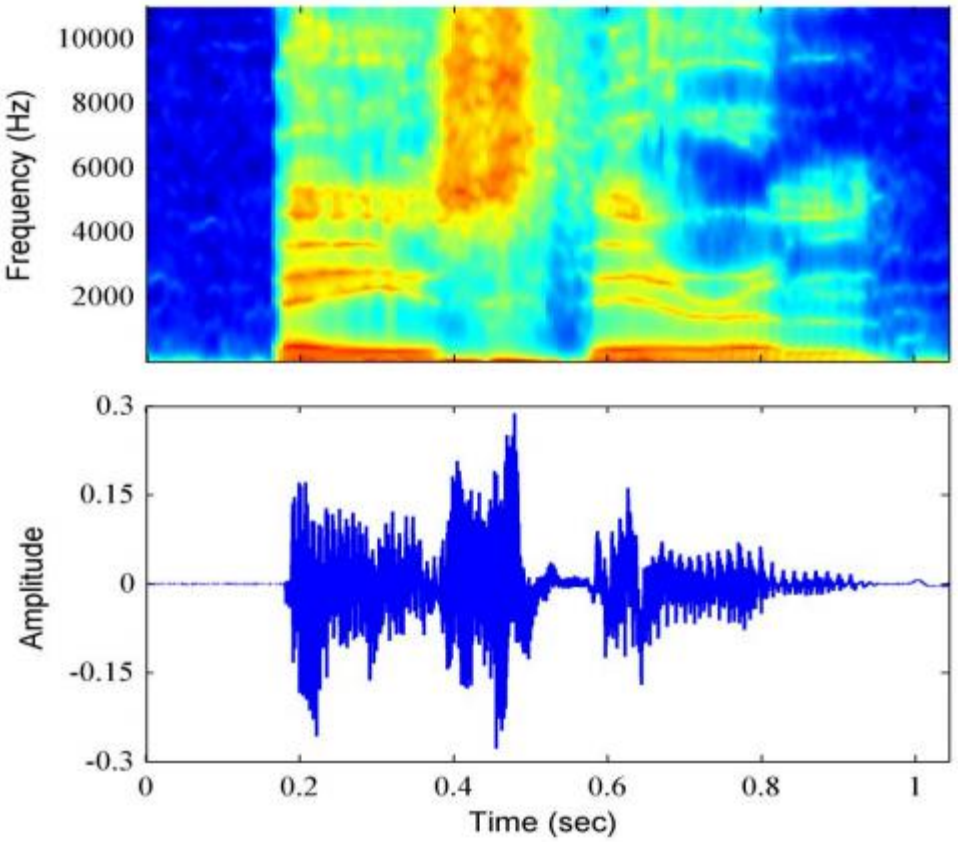
$$p(A,B) = \sum_C p(A,B,C) = \sum_C p(A)p(B)p(C|A,B) = p(A)p(B)$$

Outline

- Introduction to Graphical Model
- Introduction to Belief Networks
- **Hidden Markov Models**

Hidden Markov Models

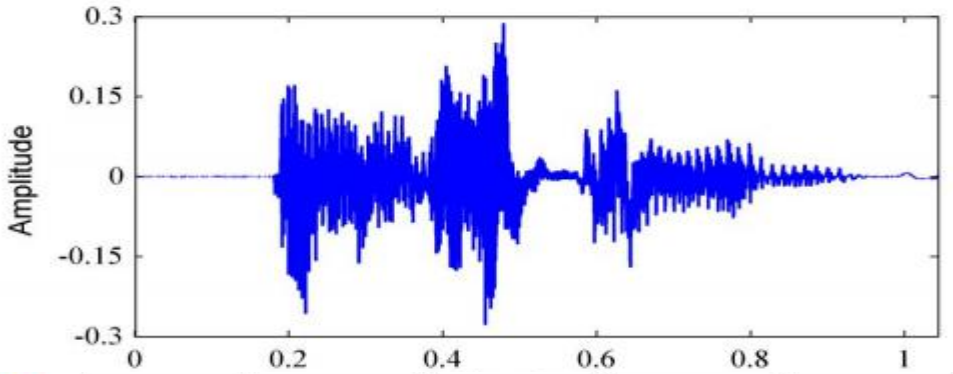
- So far we assumed independent, $\{X_i\}_{i=1}^n \overset{iid}{\sim} p(\mathbf{X})$ identically distributed data.
- Sequential data
  – Time-series data
  E.g. Speech



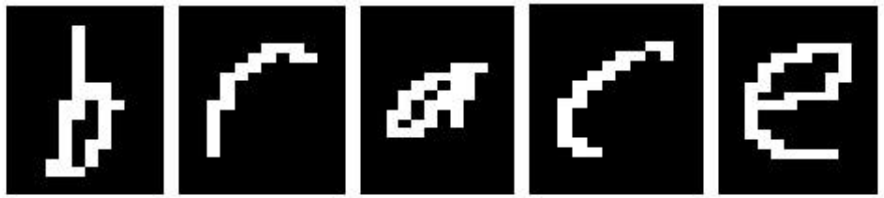
i.i.d to sequential data

- So far we assumed independent, $\{X_i\}_{i=1}^n \overset{iid}{\sim} p(\mathbf{X})$ identically distributed data.
- Sequential data
  - Time-series data

  E.g. Speech



  - Characters in a sentence



  - Base pairs along a DNA strand

Markov Models

- Joint Distribution

$$
\begin{aligned}
p(\mathbf{X}) &= p(X_1, X_2, \ldots, X_n) \\
&= p(X_1)p(X_2|X_1)p(X_3|X_2, X_1)\ldots p(X_n|X_{n-1}, ..., X_1) \\
&= \prod_{i=1}^{n} p(X_n|X_{n-1}, \ldots, X_1) \qquad \text{Chain rule}
\end{aligned}
$$

- Markov Assumption (m−th order)

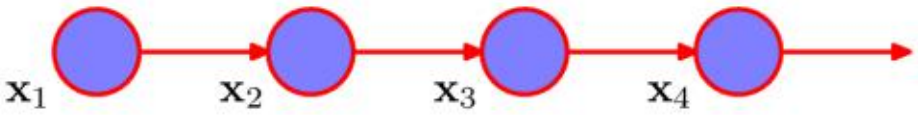
$$
p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1}, \ldots, X_{n-m})
$$

Current observation only depends on past m observations

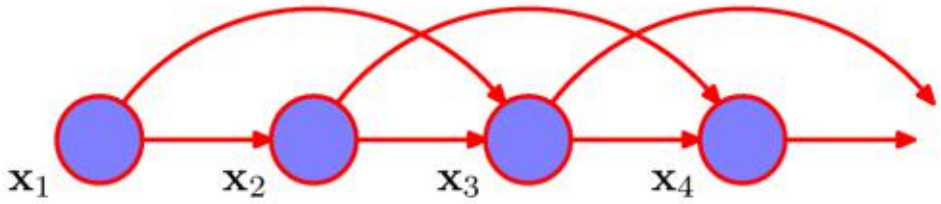
Markov Models

- 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

parameters in stationary model K-ary variables

- Markov Assumption

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

mth order $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_{n-m})$ $O(K^{m+1})$

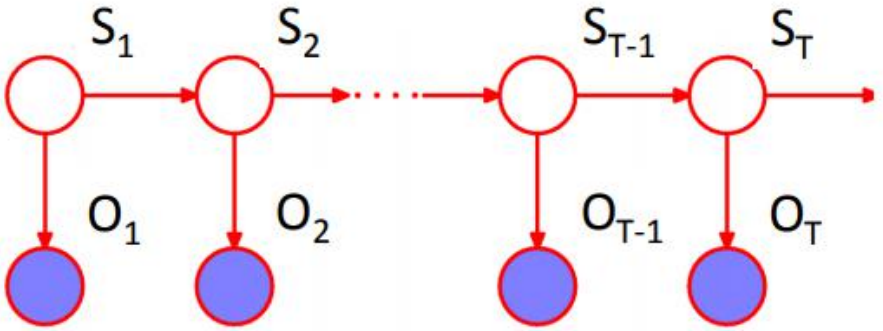
n-1th order $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_1)$ $O(K^n)$

≡ no assumptions – complete (but directed) graph

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

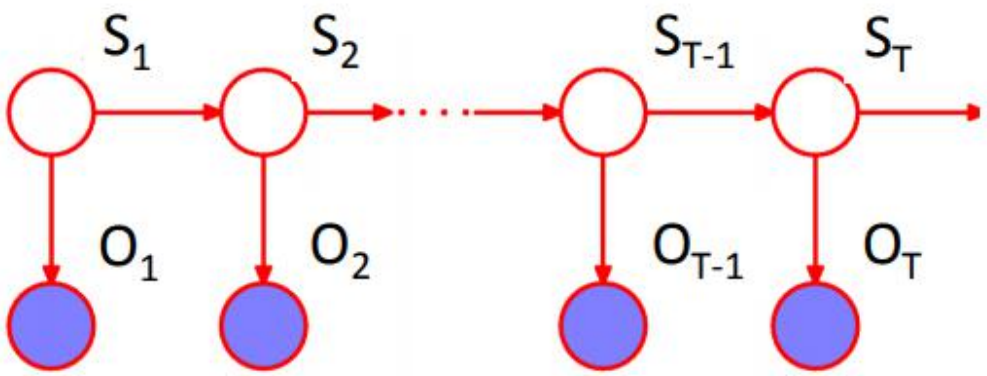
Hidden Markov Models

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



Observation space	$O_t \in \{y_1, y_2, \ldots, y_K\}$
Hidden states	$S_t \in \{1, \ldots, I\}$

Hidden Markov Models



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

- 1−st order Markov assumption on hidden states {St} t = 1, …, T (can be extended to higher order).
- Note: Ot depends on all previous observations {Ot−1,…O1}

Hidden Markov Models

- Parameters – stationary/homogeneous markov model (independent of time t)

Initial probabilities

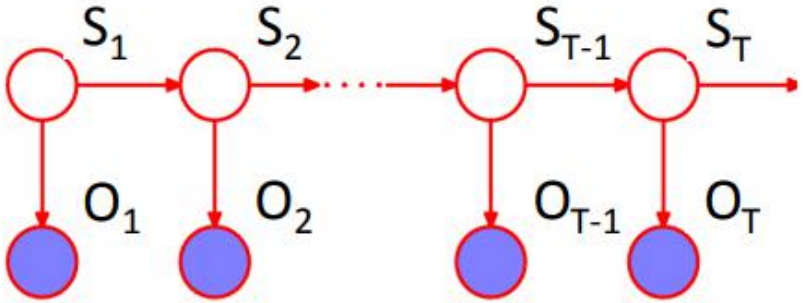
$p(S_1 = i) = \pi_i$

Transition probabilities

$p(S_t = j | S_{t-1} = i) = p_{ij}$

Emission probabilities

$p(O_t = y | S_t = i) = q_i^y$



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

HMM Example

- The Dishonest Casino

  A casino has two die:

  Fair dice

  $P(1) = P(2) = P(3) = P(5) = P(6) = 1/6$

  Loaded dice

  $P(1) = P(2) = P(3) = P(4) = P(5) = 1/10$

  $P(6) = ½$

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

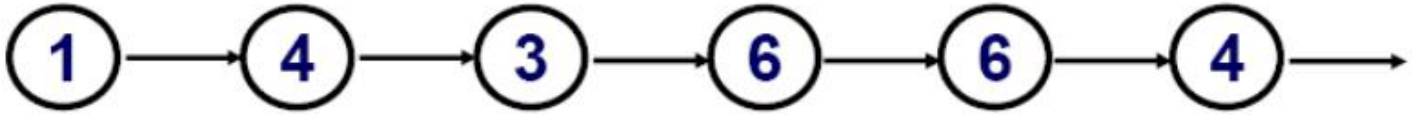
HMM Problems

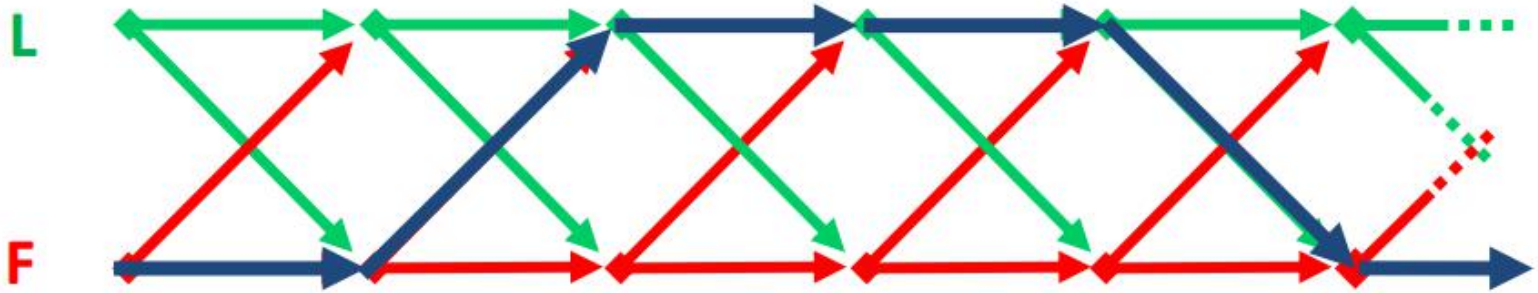
- **GIVEN**: A sequence of rolls by the casino player

- **QUESTION**
- 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 "loaded" 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

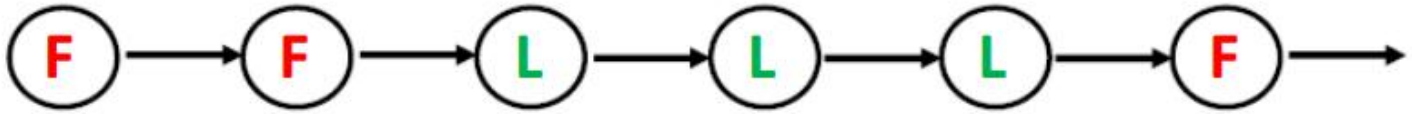
HMM Example

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



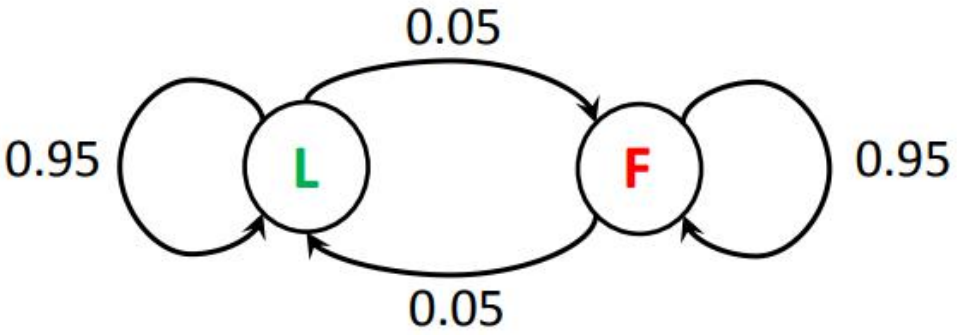


- Hidden sequence $\{S_t\}_{t=1}^{T}$ or segmentation):



State Space Representation

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



- HMM Parameters

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

Three main problems in HMMs

- **Evaluation** – Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$

  find $p(\{O_t\}_{t=1}^T)$ prob of observed sequence

- **Decoding** – Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$

  find $\arg \max_{s_1,\ldots,s_T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T)$ most probable

  sequence of hidden states

- **Learning** – Given HMM with unknown parameters and $\{O_t\}_{t=1}^T$ observation sequence

  find $\arg \max_{\theta} p(\{O_t\}_{t=1}^T | \theta)$ parameters that maximize

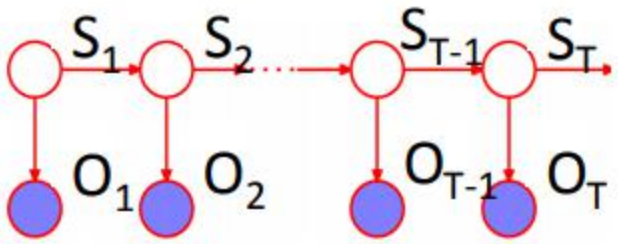
  likelihood of observed data

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 $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ and 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 number terms!

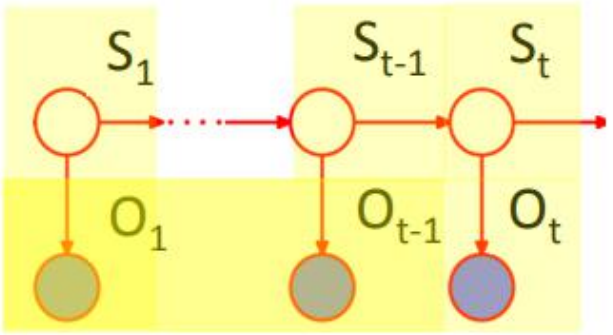
Instead: $p(\{O_t\}_{t=1}^T) = \sum_k \underbrace{p(\{O_t\}_{t=1}^T, S_T = k)}_{\alpha_T^k}$ Compute recursively

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

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



Introduce S_{t-1}

Chain rule

Markov assumption

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

Forward Algorithm

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

- Initialize: $\alpha_1^k = p(O_1 | S_1 = k)\, p(S_1 = k)$ for all k
- Iterate: for t = 2, …, T

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

- Termination: $p(\{O_t\}_{t=1}^T) = \sum_k \alpha_T^k$

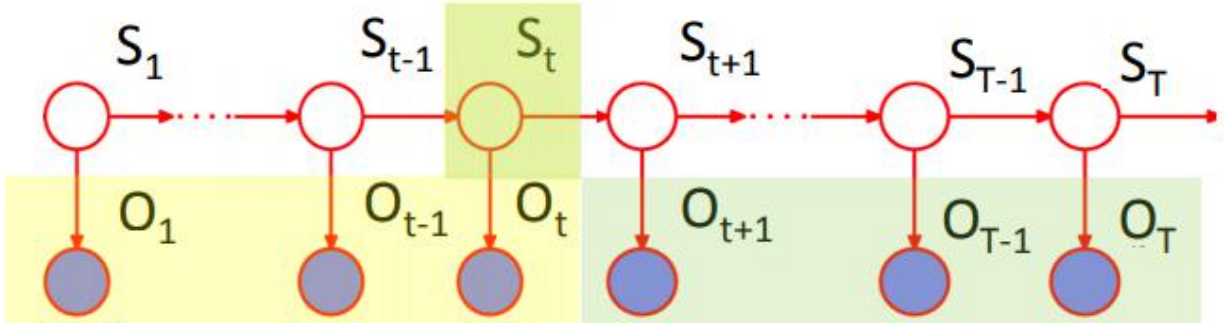
Decoding Problem 1

- Given HMM parameters $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ and 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

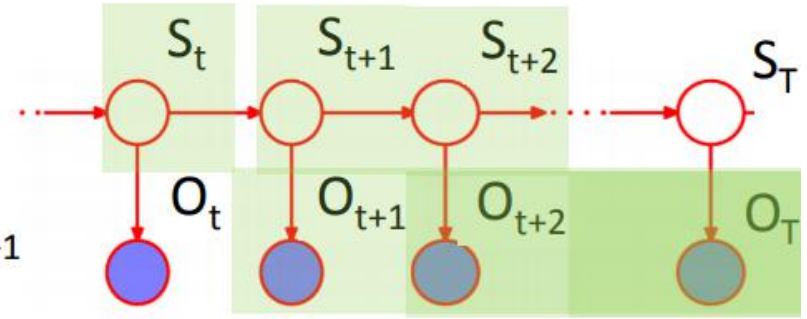


Backward Probability

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

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 β_t^k for all k, t using dynamic programming:

- Initialize: $\beta_T^k = 1$ for all k
- Iterate: for t = T-1, …, 1

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

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

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

Most likely state vs. Most likely sequence

- Most likely state assignment at time t

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

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

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

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

Not the same solution !

MLA of x?
MLA of (x,y)?

x	y	P(x,y)
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

Decoding Problem 2

- Given HMM parameters $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ and 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

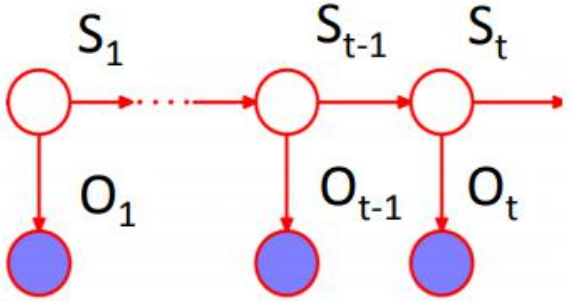
$$\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k V_T^k$$

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 for all 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 $\theta = \{\{\pi_i\}, \{p_{ij}\}, \{q_i^k\}\}$ and observation sequence $\mathbf{O} = \{O_t\}_{t=1}^T$

  find parameters that maximize likelihood of observed data

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

  But likelihood doesn't factorize since observations not i.i.d.

  hidden variables – state sequence $\{S_t\}_{t=1}^T$

  EM (Baum-Welch) Algorithm:

  E-step – Fix parameters, find expected state assignments

  M-step – Fix expected state assignments, update parameters

Baum-Welch (EM) Algorithm

- Start with random initialization of parameters
- **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

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

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

- **M-step**

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

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

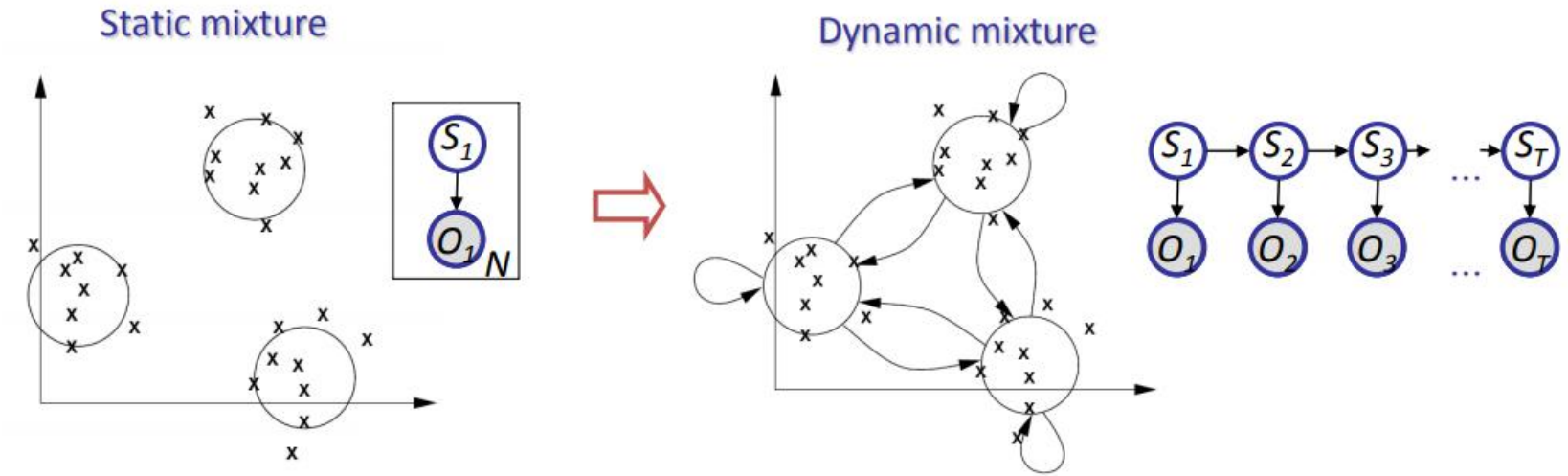
$$q_i^k = \frac{\sum_{t=1}^{T} \delta_{O_t=k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Some connections

- HMM & Dynamic Mixture Models

$$p(O_t) = \sum_{S_t} p(O_t|S_t)p(S_t)$$

Choice of mixture component depends on choice of components for previous observations



HMMs.. What you should know

- Useful for modeling sequential data with few parameters using discrete hidden states that satisfy Markov assumption
- Representation–initial prob, transition prob, emission prob, State space representation
- 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

Q & A