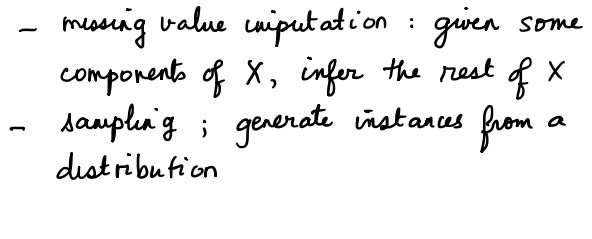
Structured Probabilistic Models

Monday, March 19, 2018 9:46 PM

For many tasks (other than classification), a full representation of the probability distribution on X or the joint distribution XY is required.

- density estimation $P(X)$
- denoising : find X which is a cleaned up version of observed $\widetilde{X}$
- missing value imputation : given some components of X, infer the rest of X
- sampling ; generate instances from a distribution

Example : Figure 16.1 on page 552 of your textbook shows synthesized images on a probabilistic model trained on CIFAR-10 data set.



Modeling a distribution on thousands or millions of random variables is a challenging task. For a small $32 \times 32 \times 3$ pixel image, there are 2^{3072} binary images of this form. 10^{800} times the estimated number of atoms in the universe!

In general, to model a random vector X containing n discrete variables, each of which can take on k values, a naive approach of using a lookup table to store P(X) will require k^n parameters.
A table based approach is not practical
- exponential cost of storage
- cost of inference is exponential
- cost of sampling is exponential.

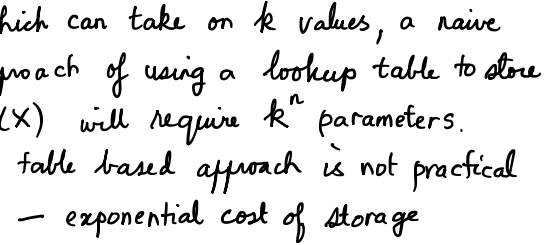
Table based approach allows us to model every possible kind of interaction between every possible subset of variables. But in most cases of distributions encountered in real world problems, most variables influence each other only indirectly.

Direct and indirect interaction

Alice $\xrightarrow{\text{baton}}$ Bob $\xrightarrow{\text{baton}}$ Carol relay race.

Finishing times t_A, t_B, t_C
- t_B depends directly on t_A
- t_C depends directly on t_B
- t_C depends indirectly on t_A
- t_A does not depend on t_B or t_C

This structure of dependencies can be represented by a directed acyclic graph. Vertices are random variables, edges denote direct dependencies



This graph is called a Bayesian network. Graph has associated set of local conditional probability distributions

$$P(x_i \mid \text{Parents}(x_i))$$

where x_i is a node in the graph.
The joint probability over all variables in the graph x

$$P(x) = \prod_i P(x_i \mid \text{Parents}(x_i))$$

$$P(t_A, t_B, t_C) = P(t_A) P(t_B \mid t_A) P(t_C \mid t_B)$$

- graph represents a factoring of the full joint distribution.

Suppose t_A, t_B, t_C are discrete with 100 values (minute 0 to minute 10 divided into 6 second chunks)

To represent full joint distribution on t_A, t_B, t_C : $10^6 - 1$ values

To represent $P(t_A)$: $10^2 - 1$ values
$P(t_B \mid t_A)$: $10^2(10^2 - 1)$ values
$P(t_C \mid t_B)$: $10^2(10^2 - 1)$ values
19,899 values

Using the directed graphical model reduced our number of parameters by a factor of more than 50!

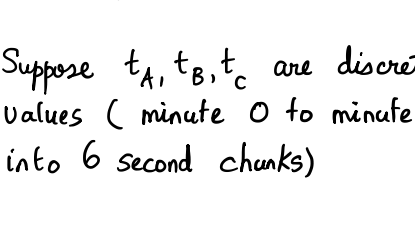
$O(k^n) \rightarrow O(k^m)$ if each variable in graph has no more than m parents

Undirected models

- Markov networks or Markov random fields
- undirected graphs

Directed models are appropriate for domains where there is clear causality at work. When interactions have no directionality, undirected models are best.

A model over 3 variables
h_y : boolean ; are you sick
h_r : boolean ; is your roommate sick
h_c : boolean ; is your coworker sick



Variables that directly interact with each other are connected by an edge.

$$\tilde{P}(x) = \prod_{C \in G} \phi(C)$$

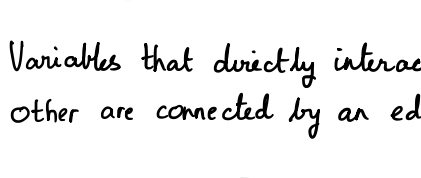
where C is a clique in the graph G. For each clique C in G, we define a factor $\phi(C)$ or clique potential, which measures the affinity for being in each of their joint states.

$\tilde{P}(x)$ is an unnormalized probability distribution.

good health.

	$h_y = 0$	$h_y = 1$
$h_c = 0$	2	1
$h_c = 1$	1	10

	$h_y = 0$	$h_y = 1$
$h_r = 0$	2	1
$h_r = 1$	1	10



$$\tilde{P}(h_r, h_y, h_c) = \phi_1(h_y, h_c)\, \phi_2(h_y, h_r)$$

$\phi(C) \geq 0$
Product of $\phi(C)$'s may not sum to 1.

h_r	h_y	h_c	$\phi_1(h_y, h_c)$	$\phi_2(h_y, h_r)$	$\tilde{P}$
0	0	0	2	2	4
0	0	1	2	1	2
0	1	0	1	1	1
0	1	1	1	10	10
1	0	0	1	2	2
1	0	1	1	1	1
1	1	0	10	1	10
1	1	1	10	10	100

We introduce a normalization factor Z

$$P(X) = \frac{1}{Z} \tilde{P}(X)$$

where $Z = \int_x \tilde{P}(x)\, dx$ or $\sum_x \tilde{P}(x)$
partition function

$Z = 4 + 2 + 1 + 10 + 2 + 1 + 10 + 100 = 130$

Now we can convert $\tilde{P}(h_r, h_y, h_c)$ into a probability by dividing it by Z (which is 130 for this example)

Need to compute Z puts restrictions on the form of clique potentials ϕ.

Cliques, factors and energy

Cliques in the undirected graph correspond to factors in the unnormalized probability function.

If X is the set of nodes in an undirected graph G,

$$\left.\begin{array}{l} \forall x \ \tilde{P}(x) > 0 \\ \tilde{P}(x) = \prod_{C \in G} \phi(C) \end{array}\right\}$$

where $C \subseteq X$ is a clique of G.

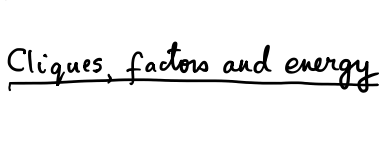
It is easy to meet these requirements if

$$\tilde{P}(x) = \exp(-E(x))$$

high energy ⟷ low $\tilde{P}$
low energy ⟷ high $\tilde{P}$

energy function
Boltzmann distribution

- $\exp(z) > 0$ for all z.



$\phi(h_r, h_y) = \exp(-E(h_r, h_y))$
$\phi(h_y, h_c) = \exp(-E(h_y, h_c))$

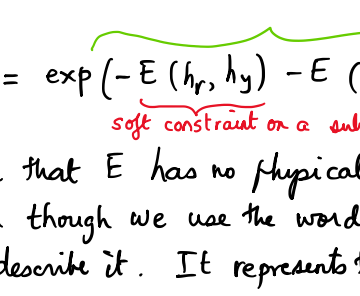
$$\phi(h_r, h_y, h_c) = \phi(h_r, h_y) \cdot \phi(h_y, h_c)$$
$$= \exp(-E(h_r, h_y) - E(h_y, h_c))$$

So $\tilde{P}(x) = \phi(x) = \phi(h_r, h_y, h_c)$

$$= \exp\Big(\overbrace{\underbrace{-E(h_r, h_y)}_{\text{soft constraint on a subset of variables}} - E(h_y, h_c)}^{\text{global const. on all vars}}\Big)$$

Note that E has no physical meaning, even though we use the word "energy" to describe it. It represents the affinity of the variables in the clique having a given joint combination of values

Example



$$P(a,b,c,d,e,f) = \frac{1}{Z} \phi_1(a,b)\, \phi_2(a,d)\, \phi_3(b,e)\, \phi_4(b,c)\, \phi_5(e,f)$$

With an energy based model where each ϕ has the form
$$\phi(x,y) = \exp(-E(x,y))$$

$$P(a,b,c,d,e,f) = \frac{1}{Z} \exp(-E(a,b)) * \exp(-E(a,d)) * \exp(-E(b,e)) * \exp(-E(b,c)) * \exp(-E(e,f))$$

$$= \frac{1}{Z} \exp\left(-\left[E(a,b) + E(a,d) + E(b,e) + E(b,c) + E(e,f)\right]\right)$$

Representational theorem

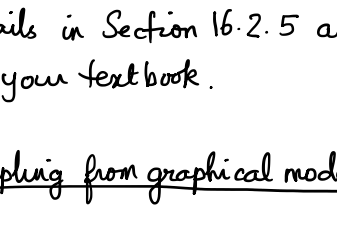
Every probability distribution can be represented by a either a directed graph model or an undirected graph model.

When we represent a probability distribution by a graph, we choose a graph that implies as many independences as possible, without implying any independences that do not hold for that domain or problem.

Details in Section 16.2.5 and Section 16.2.6 in your textbook.

Sampling from graphical models

With directed graphical models, it is a polynomial time operation to generate a sample from the full joint distribution over the nodes in a graph.



Ancestral sampling
- Sort the variables in topological order. $x_1 \dots x_n$
e.g. A, B, C, D, E
- Sample variable x_i from the distribution $P(x_i \mid \text{Parents}(x_i))$
$i = 1, \dots, n$

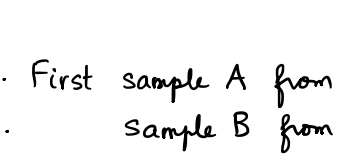
1. First sample A from P(A)
2. Sample B from P(B)
3. Sample C from P(C|A,B)
4. Sample D from P(D|C)
5. Sample E from P(E|C)

For undirected graphical models, we use Gibbs sampling. Suppose we have an undirected graphical model over variables $x_1, \dots, x_n$.

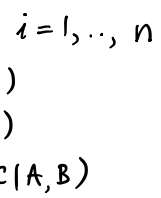
1. Start with randomly generated values for $x_1 \dots x_n$.
2. Iteratively visit each x_i and sample a value for it based on $P(x_i \mid \{x_1, \dots, x_n\} - \{x_i\})$
$\equiv P(x_i \mid \text{neighbors}(x_i) \text{ in graph})$
3. repeat 2, generating a stream of samples, where successive samples differ only on one variable.

In the long run, these samples come from $P(x_1, \dots x_n)$.

Latent variables + compact networks



59 parameters
where all vars are boolean
d_1 : 1
d_2 : 1
d_3 : 1
s_1 : 8
s_2 : 16
s_3 : 32



17 parameters
where all vars are boolean
d_1 : 1
d_2 : 1
d_3 : 1
h : 8
s_1 : 2
s_2 : 2
s_3 : 2

Introducing latent (unseen or unobserved) variables to represent a distribution on observed variables can lead to very compact representations. Latent variables can be inferred from observed variables as we will see with Hidden Markov models.
$P(h \mid d_1 d_2 d_3)$ or

$P(h \mid s_1 s_2 s_3)$ etc...

These are called inference problems and we will study them in the context of directed graphical models next.

Generative models for processes that evolve over time

Let w_t = weather at time t where $w_t \in \{$sunny, rainy, cloudy$\}$

Further let w_t be a stochastic function of w_{t-1}, weather at time t-1

graphical model which represents the evolution of weather.

$P(W_0)$: weather at time 0
$P(W_t \mid W_{t-1})$: weather at time t given weather at time t-1

$$P(W_0) = [1 \quad 0 \quad 0]^T$$

$$P(W_t \mid W_{t-1}) = \begin{array}{c|ccc} & S & R & C \\ \hline S & 0.5 & 0.25 & 0.25 \\ R & 0.5 & 0 & 0.5 \\ C & 0.25 & 0.25 & 0.5 \end{array} = q$$

transition matrix

What can be done with this graphical model?

(1) Prediction: $P(W_1)$

$$P(W_1 = x) = \sum_{x'} P(W_1 = x \mid W_0 = x')\, P(W_0 = x')$$

For $P(W_0) = [1, 0, 0]^T$
we can derive the fact that

$$P(W_1) = [0.5 \quad 0.25 \quad 0.25]^T$$
$$= [1, 0, 0]^T * q$$

In general,

$$P(W_t) = P(W_0) * q^t$$

Surprisingly, there exists a p^* such that

$$p^* = p^* q$$

p^* is the fixed point of transition matrix q.
For this q, $p^* = [0.4, 0.2, 0.4]^T$

(2) generate samples from model

- pick W_0 from $P(W_0)$
- for t = 1 to T
  pick W_t from $q(W_{t-1}, :)$

This graphical model is called a Markov chain. It is a generative model of sequences.

Here is a generative model that produces random DNA sequences.

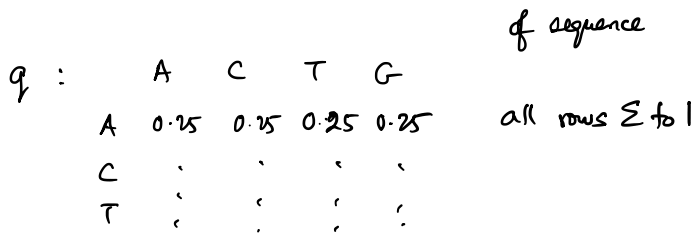
$S = \{A, C, T, G\}$ state set
$\Pi = [0.25, 0.25, 0.25, 0.25]^T$: probability of initial element of sequence

$$q : \begin{array}{c|cccc} & A & C & T & G \\ A & 0.25 & 0.25 & 0.25 & 0.25 \\ C & \vdots & \vdots & \vdots & \vdots \\ T & \vdots & \vdots & \vdots & \vdots \\ G & 0.25 & 0.25 & 0.25 & 0.25 \end{array}$$

all rows Σ to 1

Hidden Markov Models

appropriate for problems where you get a sequence of observations from a process, and you need to infer a underlying latent variable from that observation sequence.



Examples:

(1) Speech recognition: $e_1 \dots e_t$: speech
$X_0 X_1 \dots X_t$: phonemes

(2) Robot localization: $e_1 \dots e_t$: GPS readings
$X_0 X_1 \dots X_t$: location and orientation of robot

(3) Tracking health : $e_1 \dots e_t$ BP readings
$X_0 X_1 \dots X_t$ healthy or not (hypertension)

The dishonest casino example

Hidden states $S = \{F, L\}$ fair or loaded coin
Observations $O = \{h, t\}$ heads, tails

Initial state probability $\Pi : [1, 0]$
$P(X_0 = F) = 1$
$P(X_0 = L) = 0$
Hidden state transition probabilities a

$$a : \begin{array}{c|cc} & F & L \\ F & 0.95 & 0.05 \\ L & 0.1 & 0.9 \end{array}$$

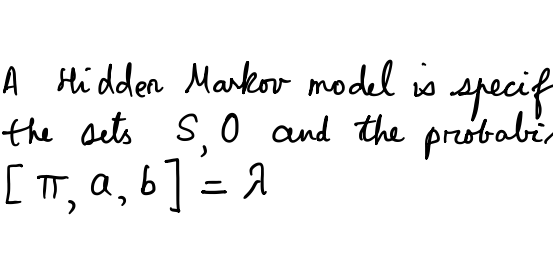
Emission probabilities b.

$$b : \begin{array}{c|cc} & h & t \\ F & 0.5 & 0.5 \\ L & 0.1 & 0.9 \end{array}$$

A Hidden Markov model is specified by the sets S, O and the probability parameters $[\Pi, a, b] = \lambda$

Given an HMM specified by S, O, λ you can generate paired sequences of observations and hidden state sequences.

(1) Sample X_0 from $P(X_0)$ (Π) ;
(2) Sample X_t from $P(X_t \mid X_{t-1})$ (a)
(3) Sample e_t from $P(e_t \mid X_t)$ (b)
repeat for t=1..T



Inference problems with HMMs

Filtering : $P(X_t \mid e_1 \dots e_t)$
Smoothing : $P(X_k \mid e_1 \dots e_t)$ $k < t$

Most likely state sequence :
$$\underset{X_1 \dots X_t}{\operatorname{argmax}} P(X_1, \dots X_t \mid e_1 \dots e_t)$$

Filtering

You see observation sequence h h h t t
What is $P(X_5 \mid h h h t t)$?

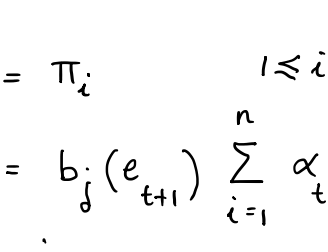
Assume you know S, O, λ (Π, a, b)

Define $\alpha_t(i) = P(e_1 \dots e_t, X_t = s_i)$

Probability of observing sequence $e_1 \dots e_t$ and X_t being $s_i \in S$.

(1) $\alpha_0(i) = \pi_i$ $\quad 1 \leq i \leq n \ ; \ |S| = n$

(2) $\alpha_{t+1}(j) = b_j(e_{t+1}) \sum_{i=1}^{n} \alpha_t(i)\, a_{ij}$ $\quad 1 \leq j \leq n$, $0 \leq t \leq T-1$



Given an observation sequence of length T, compute $\alpha_T(i)$ for $i = 1, \dots, n$.

Time complexity : $O(n^2 T)$

Smoothing

$P(X_k \mid e_1 \dots e_t)$ where $k < t$

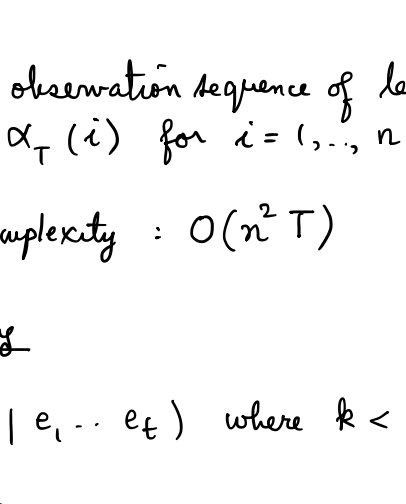
Note that
$$P(X_k \mid e_1 \dots e_t) \propto \underbrace{P(e_{k+1} \dots e_t \mid X_k)}_{\text{computed by backward computation}} \underbrace{P(X_k \mid e_1 \dots e_t)}_{\text{computed by filtering}}$$

Define $\beta_k(i) = P(e_{k+1} \dots e_t \mid X_k = s_i)$

Probability of observing $e_{k+1} \dots e_t$ assuming $X_k = s_i$.

(1) $\beta_T(i) = 1$ $\quad 1 \leq i \leq n$

(2) $\beta_k(i) = \sum_{j=1}^{n} a_{ij}\, b_j(e_{t+1})\, \beta_{k+1}(j)$ $\quad 1 \leq i \leq n$, $T-1 \leq k \leq 1$



Time Complexity : $O(n^2 T)$

Most likely state sequence

Define
$$\delta_t(i) = \max_{x_1 \dots x_{t-1}} P(x_1, \dots x_{t-1}, X_t = s_i, e_1, \dots, e_t)$$

(1) $\delta_0(i) = \pi(i)$; $1 \leq i \leq n$

(2) $\delta_{t+1}(j) = \max_i \delta_t(i)\, a_{ij}\, b_j(e_{t+1})$ $\quad 0 \leq t \leq T-1$, $1 \leq j \leq n$

Time complexity : $O(n^2 T)$

Learning the parameters of an HMM

S, O, $\lambda = (\pi, a, b)$

Case 1: paired sequences

$\left.\begin{array}{l} e_1 \dots e_T \\ x_0\ x_1 \dots x_T \end{array}\right\}$ sequence pair 1
$\left.\begin{array}{l} \text{— } e \text{ —} \\ \text{— } x \text{ —} \end{array}\right\}$ sequence pair 2
⋮

Use x_0 counts in the sequences to compute MLE estimate of π

Use hidden state sequences and count $s_i \rightarrow s_j$ transitions in them.
$$\hat{a}_{ij} = \frac{\text{count of } s_i \rightarrow s_j}{\text{count of } s_i}$$

Use hidden state/observation pairs (x_t, e_t) to estimate $b_j(o_k)$ where $o_k \in O$

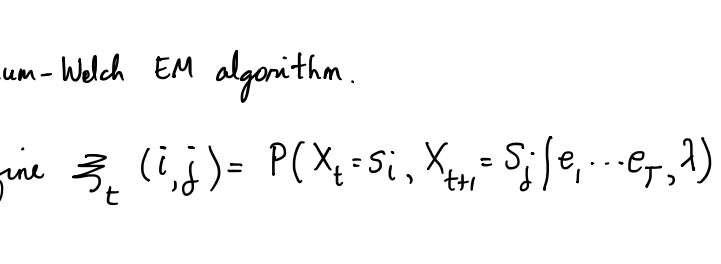
$$b_j(o_k) = \frac{\text{count of } o_k \text{ associated with } s_j}{\text{count of } s_j}$$

Case 2: observation sequence only
no hidden state sequence }

Find λ that maximizes $P(e_1 \dots e_T \mid \lambda)$

Baum-Welch EM algorithm.

Define $\xi_t(i,j) = P(X_t = s_i, X_{t+1} = s_j \mid e_1 \dots e_T, \lambda)$



$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(e_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_t(i)\, a_{ij}\, b_j(e_{t+1})\, \beta_{t+1}(j)}$$

Let $\gamma_t(i) = P(X_t = s_i \mid e_1 \dots e_T, \lambda)$
$$\gamma_t(i) = \sum_{j=1}^{n} \xi_t(i,j)$$

The expected number of times s_i is visited
$$\sum_{t=1}^{T-1} \gamma_t(i)$$

The expected number of transitions from s_i to s_j
$$\sum_{t=1}^{T-1} \xi_t(i,j)$$

Now, we can estimate
$$\hat{\pi}_i = \gamma_1(i)$$
$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$
$$\hat{b}_j(k) = \frac{\sum_{t=1,\ e_t = o_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Baum Welch Procedure

(1) Guess $\lambda_0 = [\pi_0, a_0, b_0]$
(2) Calculate α, β from λ
(3) Reestimate λ from α, β
until convergence

Demo here.