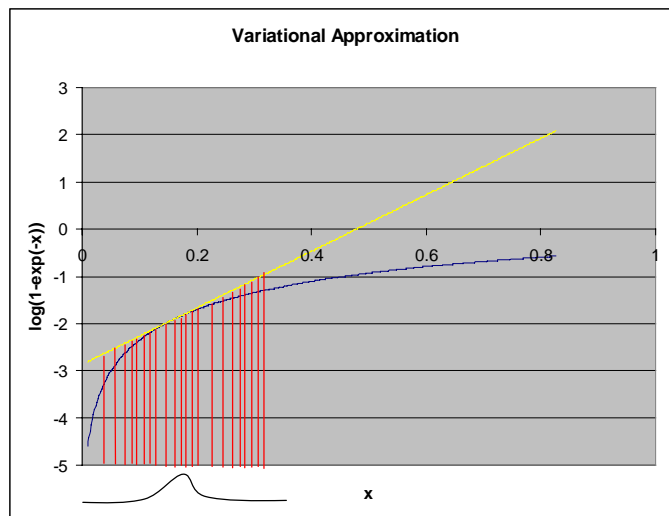


TODAY

A bit more variational approximation.
Kozlov and Koller: Dynamic Discretization



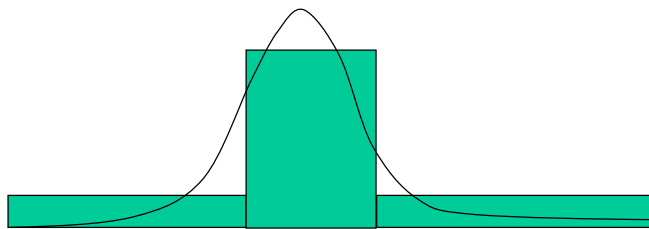
$$x = -\theta_{i_0} - \sum_{d_j \in pa(f_i)} \theta_{ij} d_j$$

Kozlov and Koller: Dynamic discretization

- Discrete approximations to continuous distributions
- Dynamic discretization:
 - Straw approach
 - Revised approach using weighted KL distance.

Discretization

Approximate a continuous probability density function by piecewise constant function:



Discretization

Assume that continuous variables are in $[0,1]$

The full state space for n variables: $\Omega = [0,1]^n$

Discretization:

A piecewise constant function $D : \Omega \rightarrow 1 \cdots m$

Defines a set of mutually exclusive and collectively exhaustive set of subregions $\{\omega_1, \dots, \omega_m\}$ in Ω

1	4	
2		5
3		

Digression: KL-distance

Why do we always use KL-distance?

Basic information theory:

Say that we want to efficiently encode a bunch of values

y_1, \dots, y_n with probabilities p_1, \dots, p_n

Shannon's encoding theorem says that the best that I can do is to code each value y_j using $\lg_2(1/p_j) = -\lg_2(p_j)$ bits.

The expected total number of bits to encode M values is

$$-M \sum_j p_j \lg(p_j)$$

and the average number of bits/word is

$$H(P) = -\sum_j p_j \lg(p_j)$$

Digression: Encoding

Say that y_1, y_2, y_3, y_4 each have probability $1/4$

the best encoding is 00, 01, 10, 11

Say that

$$P(y_1) = \frac{1}{2}, P(y_2) = \frac{1}{4}, P(y_3) = P(y_4) = \frac{1}{8}$$

the best encoding is: 0, 10, 110, 111

Digression: KL-Distance

OK, so say that we are encoding samples from P, but our coding is based on another distribution Q?

The expected number of bits to encode M letters drawn from P using a code based on Q is:

$$-M \sum_j p_j \lg(q_j)$$

If we had access to the distribution over P before encoding P, we would need only

$$-M \sum_j p_j \lg(p_j) \text{ bits.}$$

Digression: KL-Distance

The average number of “extra” bits needed to represent P if we used the wrong encoding distribution Q is:

$$\begin{aligned} KL(P\|Q) &= -\sum_j p_j \lg(q_j) - \sum_j p_j \lg(p_j) \\ &= \sum_j p_j \lg\left(\frac{p_j}{q_j}\right) \end{aligned}$$

$$KL(p\|q) = \int p(x) \ln\left(\frac{p(x)}{q(x)}\right) dx$$

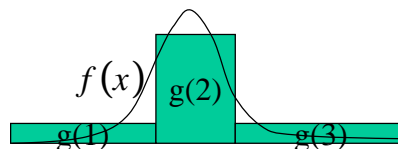
(END DIGRESSION)

Discretization

If a probability function f_D is constant in each of the subregions of discretization D, we will call it a discretized function on D.

$$f_D(x_1, \dots, x_n) = g(D(x_1, \dots, x_n))$$

How should I choose g to minimize $KL(f\|f_D)$???



Optimal values for discretization

The g that minimizes the KL distance is just:

$$g(i) = \int_{\omega_i} f(x) dx$$

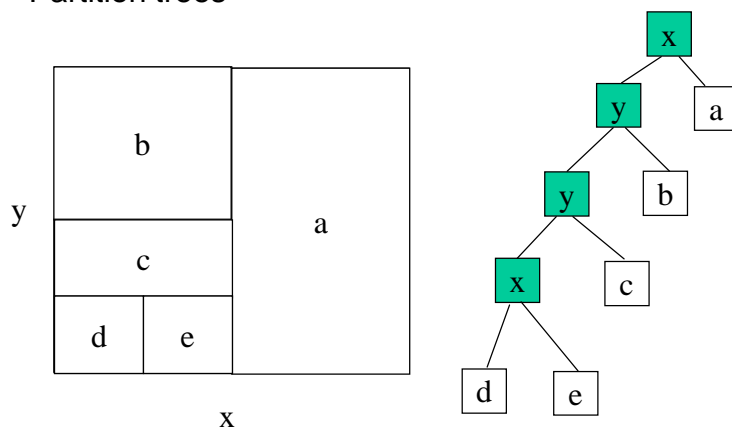
that is, g is the average value of f over each discretization region.

The KL distance to any other piecewise constant function on D is given by the sum of the KL distances:

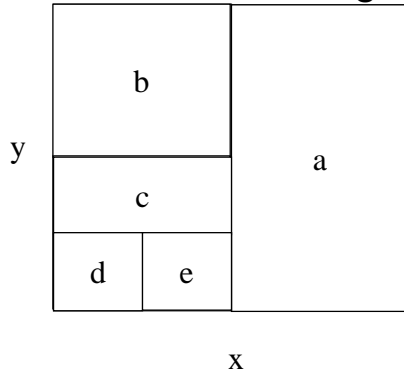
$$KL(h_D \| f) = KL(h_D \| f_D) + KL(f_D \| f)$$

How do we generate discretizations?

Consider only discretizations given by Binary Split Partition trees



Heuristic for generating good BSPs



Find leaf i with the maximum KL-distance

$$\int_{\omega_i} f(x) \ln \left(\frac{f(x)}{g(i)} \right) = \int_{\omega_i} f(x) \ln \left(\frac{f(x)}{\bar{f}(i)} \right)$$

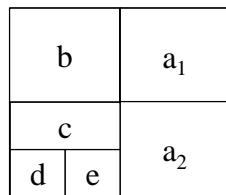
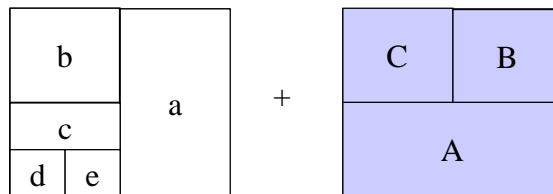
Too hard to integrate, so bound:

$$\int_{\omega_i} f(x) \ln \left(\frac{f(x)}{\bar{f}(i)} \right) \leq \left[\frac{(f_{\max} - \bar{f})f_{\min} \log(f_{\min}) + (\bar{f} - f_{\min})f_{\max} \log(f_{\max}) - (f_{\max} - f_{\min})\bar{f} \log(\bar{f})}{f_{\max} - f_{\min}} \right] |\omega_i|$$

Remember to show viewgraph with discretizations.

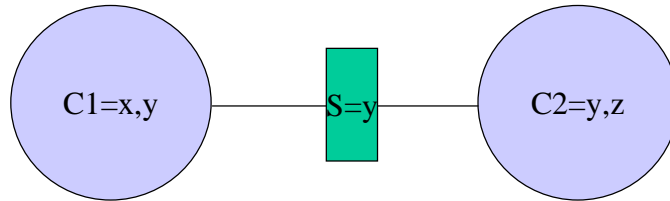
Operations on discretized functions.

Need to align the discretizations for both trees.



Values for leaves are sums of corresponding leaves for both trees.

One possible join tree algorithm.



1. Discretize C1 and C2
2. Absorb from C1 to C2
3. Absorb from C1 back to C2

$$\phi_s'(y) = \int_x \phi_c(x, y) dx$$

$$\phi_s''(y) = \int_z \phi_c'(y, z) dz$$

$$\phi_c'(y, z) = \phi_c(y, z) \frac{\phi_s'(y)}{\phi_s(y)}$$

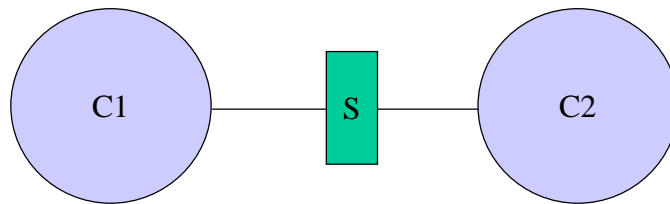
$$\phi_c'(x, y) = \phi_c(x, y) \frac{\phi_s''(y)}{\phi_s'(y)}$$

Absorption ensures that C1 and C2 are *locally consistent*:

$$\int_x \phi_c(x, y) dx = \int_z \phi_c(y, z) dz$$

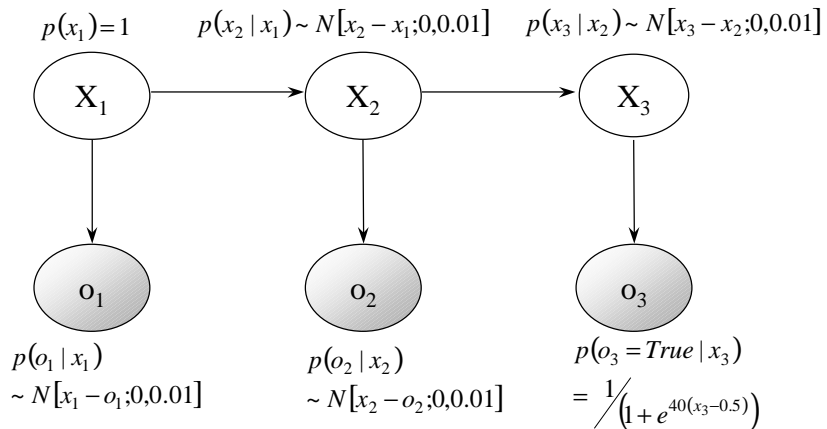
Better to inform discretization on previous observations.

[Call this “propagation with discretization”]

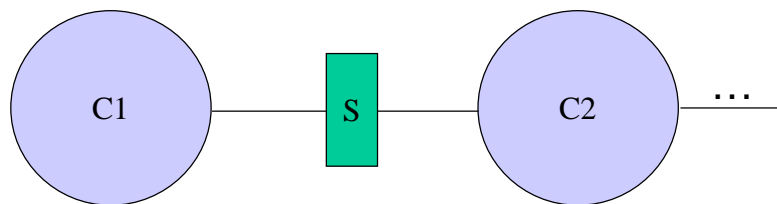


1. Discretize C1
2. Marginalize C1 to S
3. Discretize C2 * S.

Example



The problem...



Say that we discretize C2, and then propagate to C1.
 The discretization for C1 (the root) is pretty good, because it includes all of the "messages" from all of the other evidence.
 The discretization for C2 (the leaf), however, does not include the message from C1. If the prior for C2 and the likelihood due to the message from C1 are very different (low $P(E)$), the discretization is poor.

Remember example.

Dynamic Discretization

1. Use *weighted KL distance* to select the discretization.

$$WKL(p\|q; w) = \int w(x)p(x)\ln\left(\frac{p(x)}{q(x)}\right)dx \quad w(x) > 0$$

2. Goal is to assign weights to cliques so that minimizing the WKL distance minimizes the error for the probability of the query node q given evidence e .

Selecting the weights

Assume that the weight for a parent clique is already known.

How do we select the weight for a child clique to minimize the WKL distance for the parent.

The weight for the root clique is 1.

Proof: The weight functions for adjacent cliques must obey:

$$\int w(x, y)\phi_c(x, y)dx = \int w(y, z)\phi_c(y, z)dz$$

Weight Propagation

This last relationship looks like local consistency...

$$\int w(x, y) \phi_c(x, y) dx = \int w(y, z) \phi_c(y, z) dz$$

So, update weights going down the tree by using absorption to compute the product $w\phi_c$

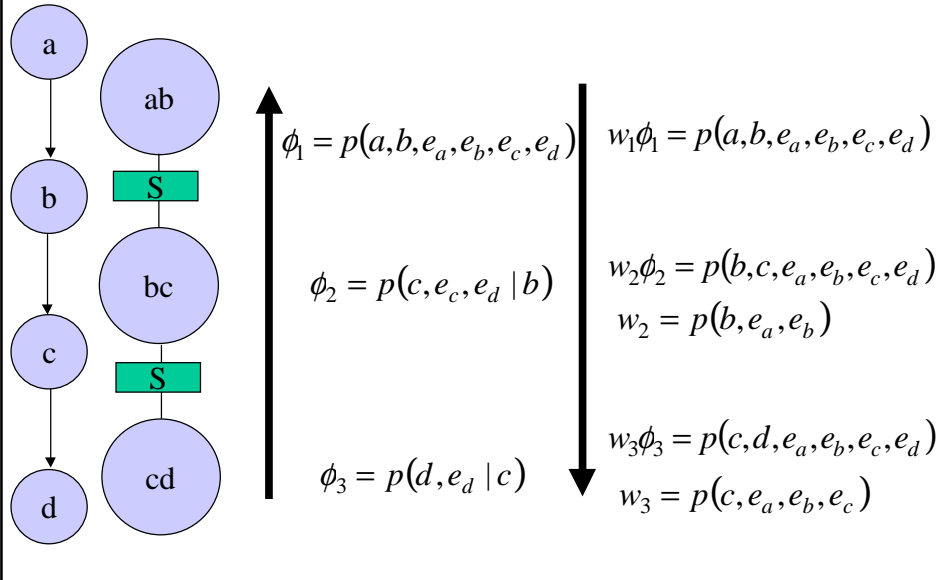
The algorithm

Iteratively determine weights.

1. Assign a weight of 1 to all cliques.
2. Use standard propagation to update all of the messages toward the root (goal) clique.
3. Using the resulting clique potentials, determine new weights by propagating the product of the weight and the clique potential back down the tree.

Repeat 2 and 3.

If discretization is perfect...



Next Time

Next week:

Skiing

The two weeks after that:

Normal distributions.

Junction tree algorithm.

Learning distributions (2 lectures).

Conjugate distributions.

Plates

Learning structure (4 lectures).