Inference & Planning

Robot Learning Summer School Lisbon Portugal, 21st July 2009

Marc Toussaint
Machine Learning & Robotics Group, TU Berlin
mtoussai@cs.tu-berlin.de

Part 1: Introduction to probabilistic inference & learning

Part 2: Planning by Inference

Outline

- **Part 1: Introduction to probabilistic inference & learning**
  – probabilities, joint distributions, graphical models
  – inference, message passing
  – learning, Expectation Maximization

- **Part 2: Planning by Inference**
  – general idea of inference by planning
  – Markov Decision Processes revisited
  – Stochastic Optimal Control revisited

- **Summary & further reading**
  – brief summary
  – further reading
  – food for thought

Part 1

Introduction to probabilistic inference & learning

– actually a big topic...

→ given limited time

– try to provide minimal self-contained background

– I focus on basic concepts necessary to understand message passing & Expectation Maximization

mostly an excerpt from this course:

https://ml01.zrz.tu-berlin.de/wiki/Main/SS09_GraphicalModels

Probability theory

- why do we need probabilities?
  – of course, in case of random events, stochasticity...

- but also in a deterministic world!:
  – partial knowledge!
  – hidden (latent) variables
  – expressing *uncertainty*
  – expressing *information*

- we use probability distributions as a generic tool to express uncertainty, information, and coupling
  – data is information
  – sensors give information
  – true state/actions/decisions are *missing* information (to be 'inferred')

Probability: Frequentist and Bayesian

- Frequentist probabilities are defined in the limit of an infinite number of trials
  – *Example:* The probability of a particular coin landing heads up is 0.43

- Bayesian (subjective) probabilities quantify degrees of belief/information/uncertainty
  – *Example:* The probability of it raining tomorrow is 0.3
  – Not possible to repeat tomorrow many times

Notation: Random variables

- a *random variable* X assignes probabilties $P(X=x) \in \mathbb{R}$ to *values* $x \in \text{dom}(X)$

  The *domain* $\text{dom}(X)$ is the set possible values of a random varible (mutually exclusive and collectively exhaustive)

- notation:
  – capital letters X to denote a random variables
  – lower case letters $x \in \text{dom}(X)$ to denote a value
  – $P(X = x) \in \mathbb{R}$ to denote the mapping to probabilites

Notation: Joint distributions

- $P(X)$ defined a single-variable probability distribution over X
  discrete case: $P(X)$ is a table/vector of entries such that $\sum_X P(X) = 1$
- $P(X, Y)$ defines a joint distribution over X and Y:
  $P(X = x, Y = y)$ gives the probability that $X = x$ *and* $Y = y$.
  discrete case: $P(X, Y)$ is a table/matrix of entries such that $\sum_{X,Y} P(X, Y) = 1$

- *definitions:*
  – the *marginal* (probability) of X given $P(X, Y)$ is $P(X) = \sum_Y P(X, Y)$
  – the *conditional* (probability) of X given Y and $P(X, Y)$ is
  $P(X|Y) = \frac{P(X,Y)}{P(Y)}$

- *implications:*
  – the *product rule* $P(X, Y) = P(X|Y)\ P(Y) = P(Y|X)\ P(X)$
  – the *chain rule* $P(X_1, .., X_n) = \prod_{i=1}^{n} P(X_i|X_1, .., X_{i\text{-}1})$
  – *Bayes Rule* $P(X|Y) = \frac{P(Y|X)}{P(Y)} P(X)$

Graphical models

- graphical models are essentially a graphical notation for the *joint distribution of coupled random variables*

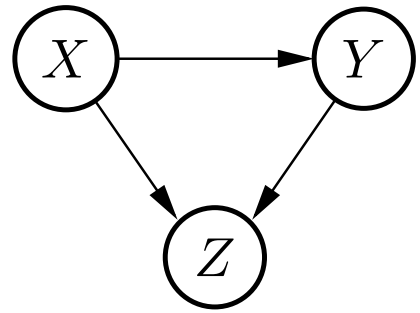
- many concrete algorithms can be derived/explained in terms of graphical models:
  – in speech & text processing (HMMs, CRFs, ..)
  – in computer vision (MRFs, sensor fusion, ..)
  – clustering (Dirichlet processes, LDA)
  – regression (GPs)
  – reinforcement learning (3rd part of lecture)
  – robotics (AICO)
  – dimension reduction (GPLVM, GTM)

- Graphical Models make things *simpler!*
  – they help to explain/unterstand many algorithms in one coherent framework
  – generic methodology to derive your own specialized algorithm

Bayesian Networks

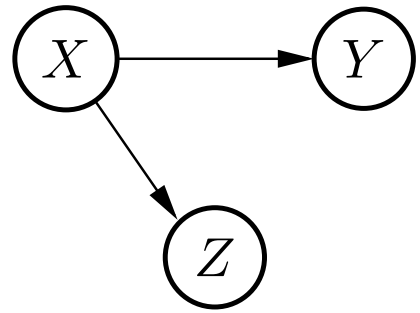
1st model:

$$P(Z, Y, X) =$$
$$P(Z|Y, X)\ P(Y|X)\ P(X)$$



2nd model:

$$P(Z, Y, X) =$$
$$P(Z|X)\ P(Y|X)\ P(X)$$



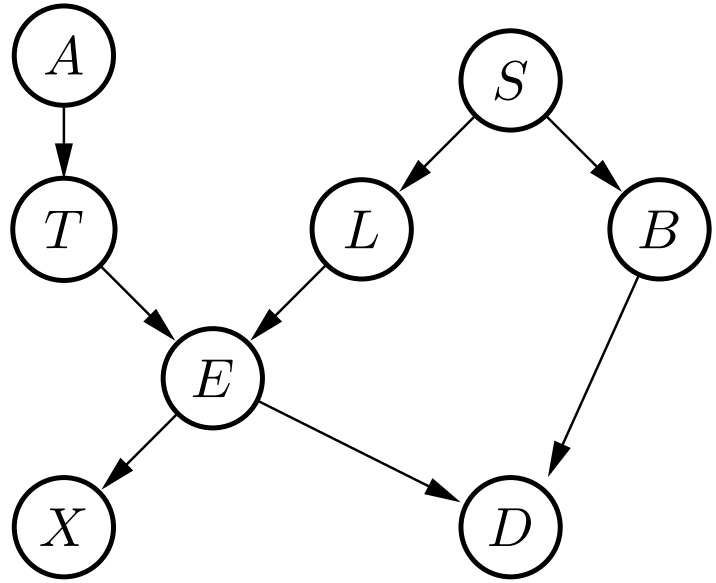
- A Bayesian network is a graphical notation of (in)dependence
- A Bayesian network is a DAG that defines for each node X_i what the parents $\pi(i)$ such that

$$P(X_{1:n}) = \prod_{i=1}^{n} P(X_i \mid X_{\pi(i)})$$

(notation: $X_{\pi(i)} = (X_a, .., X_b)$ if $\pi(i) = (a, .., b)$)

Example

the ASIA network: a model for lung disease



$$\iff P(D, X, E, B, L, T, S, A) =$$

$$P(D|E,B)\ P(X|E)\ P(E|T,L)\ P(B|S)\ P(L|S)\ P(T|A)\ P(S)\ P(A)$$

A=trip to asia

S=smoking

T=Tuberculosis

L=lung cancer

E=abnormality in chest

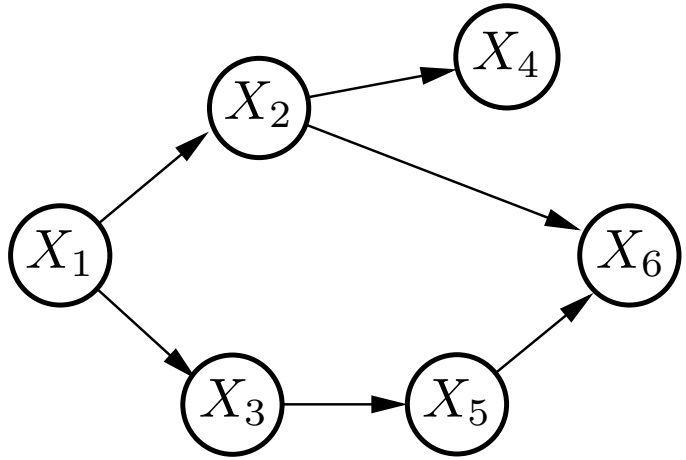
X=X-ray

D=Dyspnea

B=Bronchitis

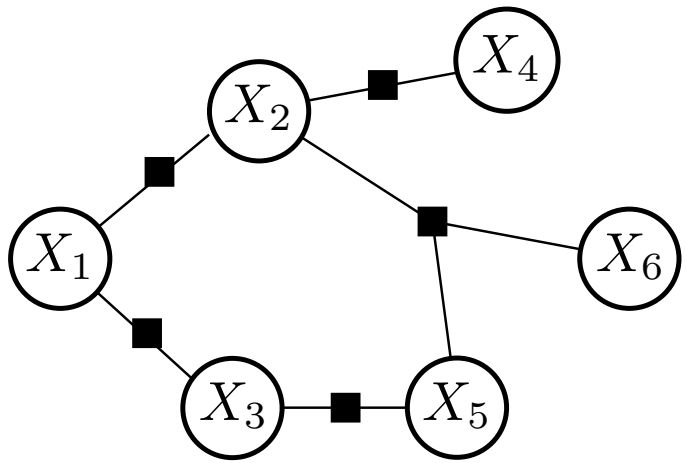
Bayes Net $\longrightarrow$ factor graph

- Bayesian Network:



$$\Longleftrightarrow \; P(X_{1:6}) = P(X_1)\; P(X_2|X_1)\; P(X_3|X_1)\; P(X_4|X_2)\; P(X_5|X_3)\; P(X_6|X_2, X_5)$$

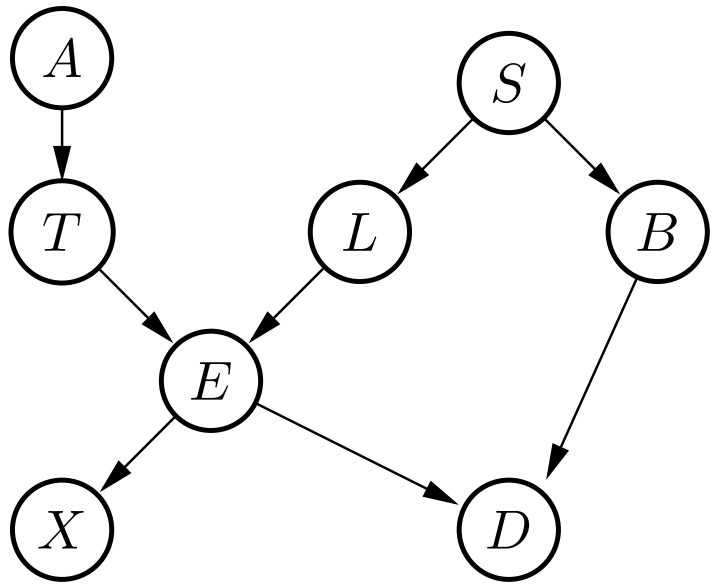
- Factor Graph: (direct notation of "how the joint factors")



$$\Longleftrightarrow \; P(X_{1:6}) = \psi_1(X_1, X_2)\; \psi_2(X_3, X_1)\; \psi_3(X_2, X_4)\; \psi_4(X_3, X_5)\; \psi_5(X_2, X_5, X_6)$$

Factor graphs

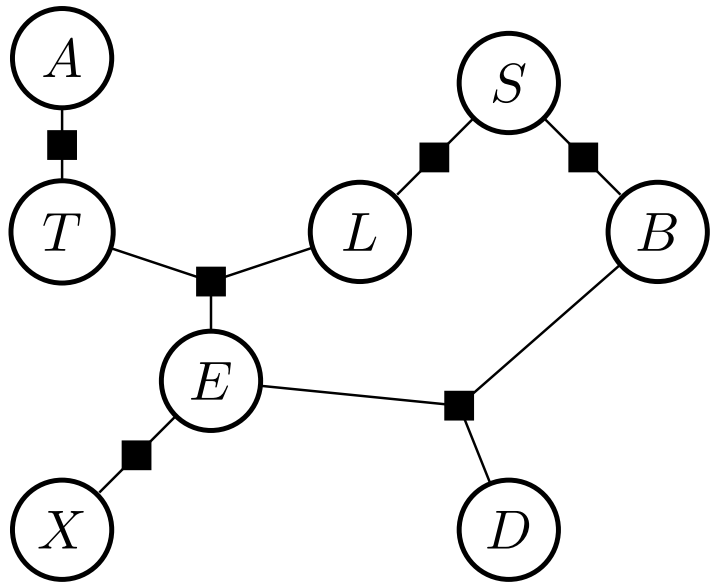
- asia example:
  Bayes Net:



$$\iff P(D, X, E, B, L, T, S, A) =$$
$$P(D|E,B)\ P(X|E)\ P(E|T,L)\ P(B|S)\ P(L|S)\ P(T|A)\ P(S)\ P(A)$$

Factor graphs

- asia example:
  factor graph: (related to *moralization* of the Bayes Net)



$$\iff P(D, X, E, B, L, T, S, A) =$$
$$\psi_1(D, E, B)\ \psi_2(X, E)\ \psi_3(E, T, L)\ \psi_4(B, S)\ \psi_5(L, S)\ \psi_6(T, A)$$

Factor graphs

- a factor graph is given by a
  – a set of random variables variables $X_1, .., X_n$
  – a set of cliques $C_1, .., C_k$ (which are tuples of variables)
  – for each clique a factor $\psi_i(X_{C_i})$ s.t.:

$$P(X_1, .., X_n) = \prod_{i=1}^{k} \psi_i(X_{C_i})$$

  (notation: $X_C = (X_a, .., X_b)$ if $C = (a, .., b)$)

- a factor graph is more general than a Bayes Net:
  – describes couplings between variables in terms of common factors
  – not only contidional probabilities

- easy to represent in a computer

Outline

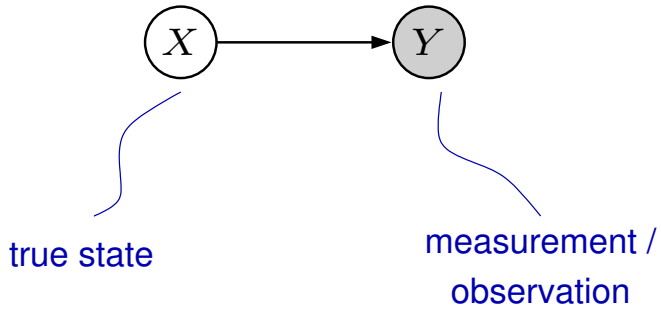
- **Part 1: Introduction to probabilistic inference & learning**
  – probabilities, joint distributions, graphical models
  – inference, message passing
  – learning, Expectation Maximization

- **Part 2: Planning by Inference**
  ...

- **Summary & further reading**
  ...

Inference

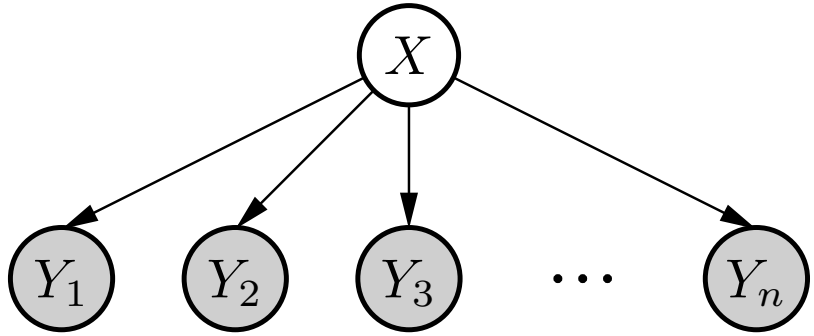
- given a joint distribution, e.g.,



  $P(X, Y) = P(Y|X)\ P(X)$
  and observation on Y

- inference $:=$ compute $P(X|Y)$

Naive Bayes



$$\iff P(X, Y_{1:n}) = P(X) \prod_{i=1}^{n} P(Y_i|X)$$

- one hidden variable, many *conditionally independent* evidences
- what is the posterior $P(X|Y_{1:n})$?

$$P(X|Y_{1:n}) := \frac{P(X, Y_{1:n})}{P(Y_{1:n})} = \frac{1}{P(Y_{1:n})} P(X) \prod_{i=1}^{n} P(Y_i \,|\, X)$$

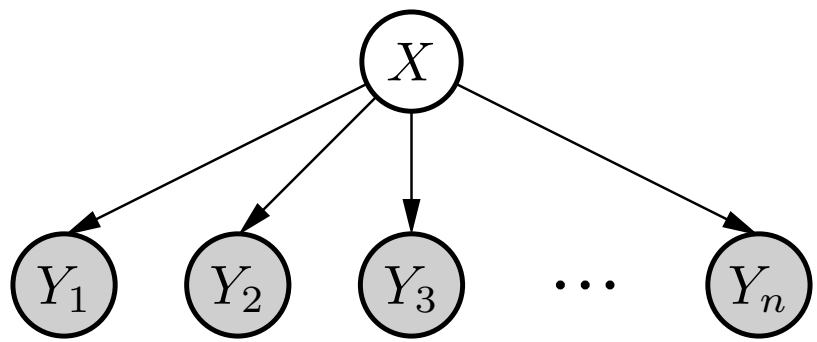
$$\propto P(X) \prod_{i=1}^{n} \mu_i(X) \,, \quad \mu_i(X) := P(Y_i = y_i \,|\, X)$$

  – the posterior is a **product** of "messages" (prob. distributions $\mu_i(X)$)

  – each independent source of information contributes a "message"

$\longrightarrow$ core operations

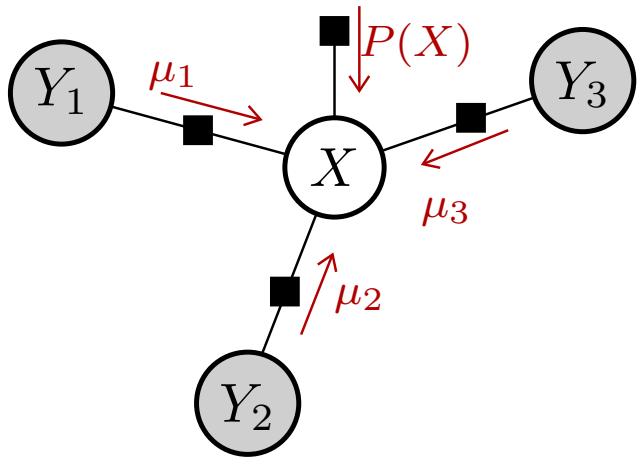
1. **summing/marginalizing**
   – marginalizes a joint distribution $P(X) = \sum_Y P(X, Y)$
   – "eliminate Y" "subsume information on Y" "resolve coupling to Y"
2. **product**
   – fusing (independent) information
   – Bayes rule $P(X|Y) \propto P(Y|X)P(X)$, posterior $\propto$ likelihood $\cdot$ prior
   – Naive Bayes



$P(X|Y_{1:n}) \propto P(X) \prod_{i=1}^{n} \mu_{Y_i \to X}(X)$ with $\mu_{Y_i \to X}(X) := P(Y_i = y_i \,|\, X)$

Message passing for Naive Bayes

- Naive Bayes as factor graph:



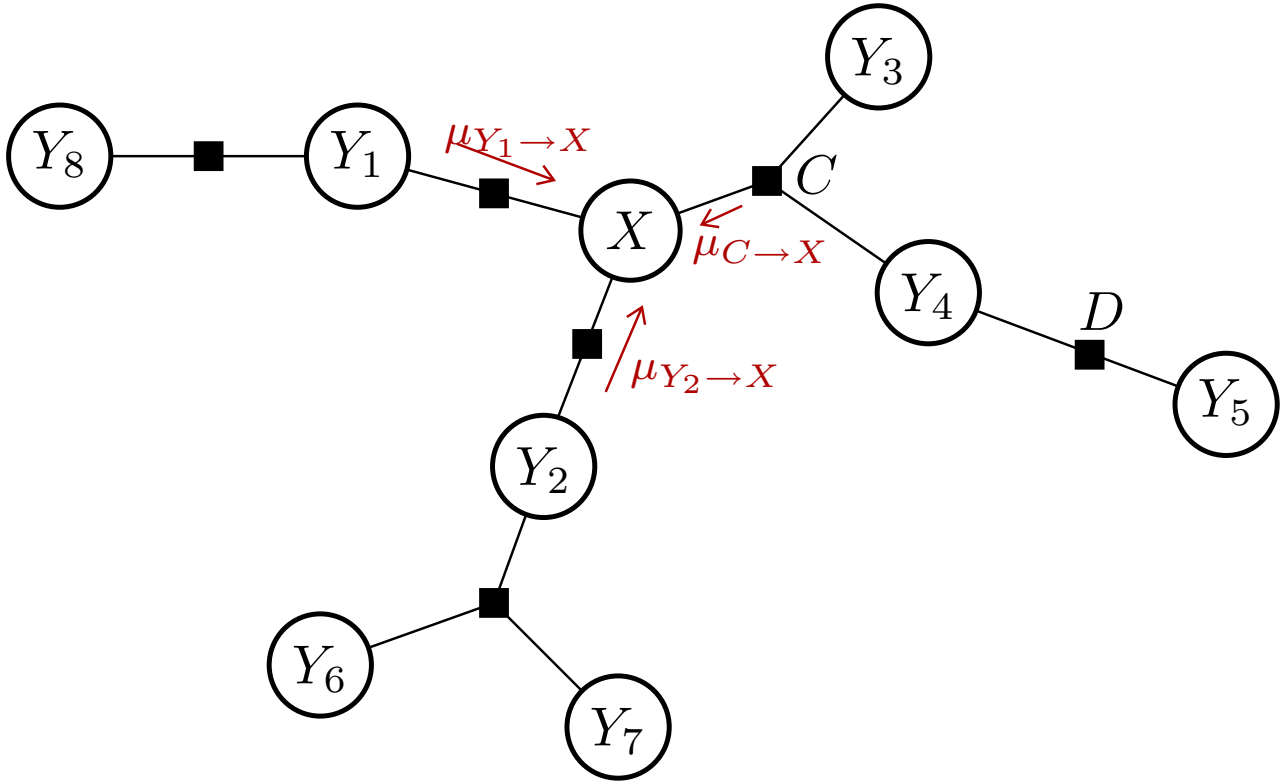
- posterior *belief* is the *product of messages*

$$P(X|Y_{1:n}) \propto P(X) \prod_{i=1}^{n} \mu_i(X)$$

  – messages are $\mu_i(X) := P(Y_i = y_i \mid X)$

Message passing on trees

- from Naive Bayes to trees:



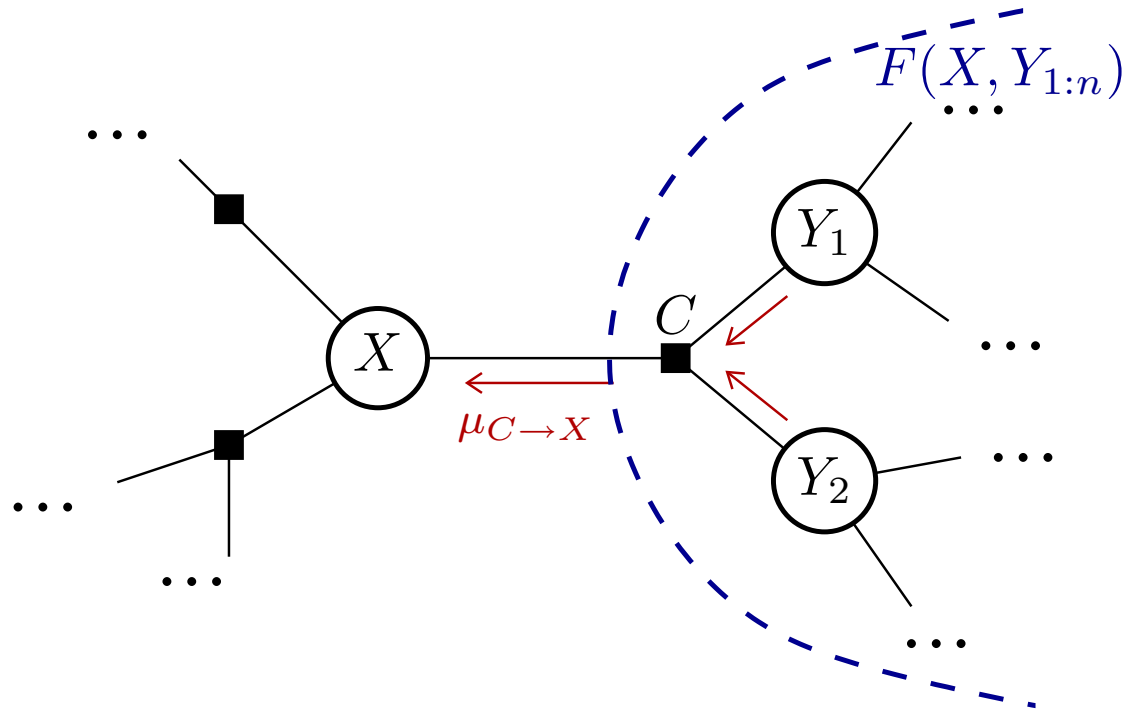
- posterior belief is still the product of messages

$$b_i(X) := \prod_{C \in \nu(X)} \mu_{C \to X}(X)$$

– messages are $\mu_{C \to X}(X) := \sum_{Y_{3,4,5}} \psi_C(X, X_3, X_4)\, \psi_D(X_4, X_5)$

Message passing

- on general tree structures: (see also Bishop: *Pattern Regocnition*)



- **messages subsume the information from a whole branch**:

  $\mu_{C\to X}(X) := \sum_{Y_{1:n}} F(X, Y_{1:n})$

- allows for a recursive computation (**message passing**):

  $\mu_{C\to X}(X) = \sum_{Y_1,Y_2} \psi_C(X, Y_1, Y_2)\ \mu_{Y_1\to C}(Y_1)\ \mu_{Y_2\to C}(Y_2)$

- the belief is the *product* of independent information:

  $b_i(X_i) = \prod_{C\in\partial i} \mu_{C\to i}(X_i)$

Message passing

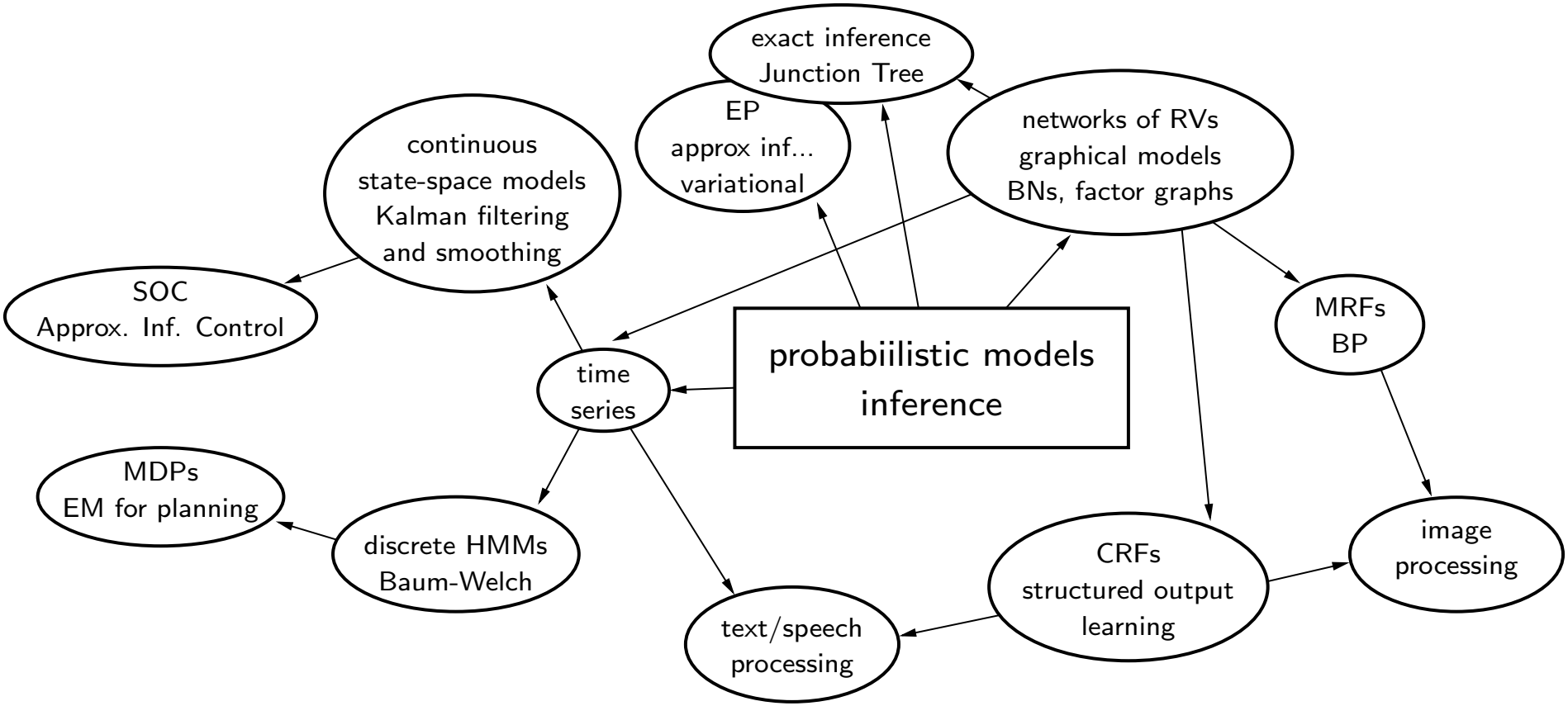
- BP can also be implemented on loopy graphs:
  1) we can't resolve recursion of msg. eqns $\rightarrow$ update eqns
  2) marginal consistency is a fixed poing of BP update eqns

$$\sum_{X_C \setminus X_i} b(X_C) = \sum_{X_D \setminus X_i} b(X_D) = b(X_i)$$

  3) problem: we multiply/fuse *dependent* information
  4) may diverge
  5) ongoing theory: Bethe approx., loop correction, generalized BP, etc

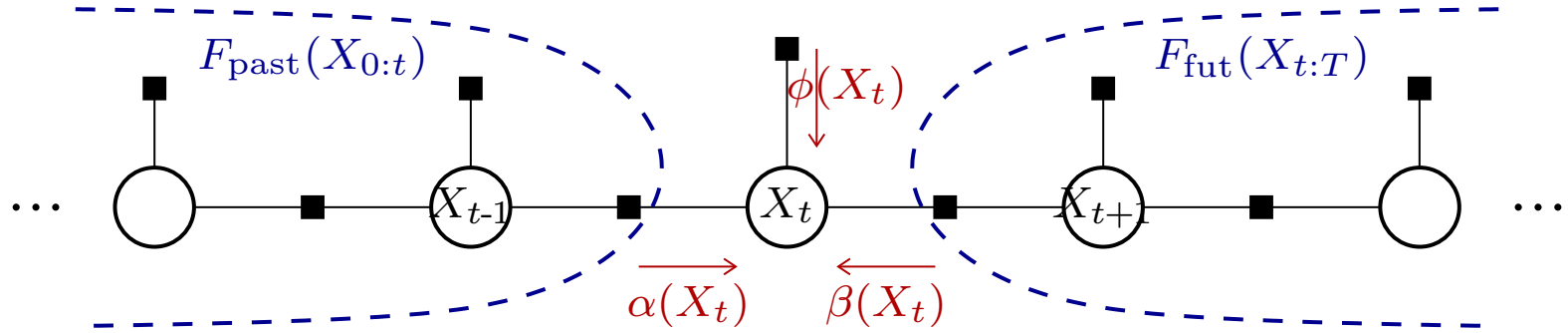
A universe of ML methods...

- many interesting directions to proceed from here...



- can't cover everything in this lecture
  we focus on time series $\rightarrow$ planning

Message passing in time series



- to compute the posterior belief at time T we need information from the past (α), the future (β), and the 'now' (ϕ)
- messages:

$$\alpha(X_T) := \sum_{X_{0:t\text{-}1}} F_{\text{past}}(X_{0:t})$$

$$= \sum_{x_{t\text{-}1}} f(X_{t\text{-}1}, X_t)\ \phi(X_{t\text{-}1})\ \alpha(X_{t\text{-}1}) \qquad \textit{forward recursion}$$

$$\beta(X_T) := \sum_{X_{t+1:T}} F_{\text{fut}}(X_{t:T})$$

$$= \sum_{x_{t+1}} f(X_t, X_{t+1})\ \phi(X_{t+1})\ \beta(X_{t+1}) \qquad \textit{backward recursion}$$

$\phi(X_T) =$ direct evidence for X_t

Message passing in time series

- given these messages, the posterior belief is the *product*

$$b(X_t) = \alpha(X_t)\ \phi(X_t)\ \beta(X_t)$$
$$b(X_t, X_{t+1}) = \alpha(X_t)\ \phi(X_t)\ f(X_t, X_{t+1})\ \phi(X_{t+1})\ \beta(X_{t+1})$$

- in discrete case:
  – model is called Hidden Markov Model (HMM)
  – inference is called “forward-backward”

- in continuous case:
  – model is called state-space model
  – lin. Gaussian: computing α’s is called “Kalman filtering”
  – lin. Gaussian: computing α’s & β’s is called “Kalman smoothing”

- that's it for inference in this lecture!

Outline

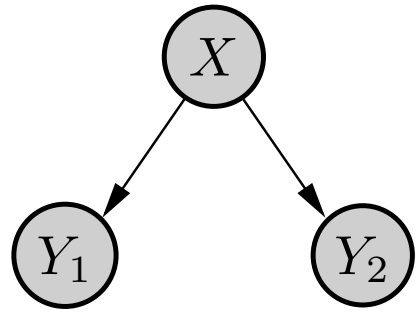
- **Part 1: Introduction to probabilistic inference & learning**
  – probabilities, joint distributions, graphical models
  – inference, message passing
  – learning, Expectation Maximization

- **Part 2: Planning by Inference**
  ...

- **Summary & further reading**
  ...

Learning with complete data

- Given three random varibles X, Y_1, Y_2 with structure



$$P(X, Y_{1,2}) = P(X)\ P(Y_1|X)\ P(Y_2|X)$$

  – unknown parameters $P(X=x) \equiv \pi_x$, $P(Y_1=y|X=x) \equiv a_{yx}$, $P(Y_2=z|X=x) \equiv b_{zx}$

  – Given a data set $\{(x_i, \boldsymbol{y}_i)\}_{i=1}^N$

  – how can we learn the parameters $\theta = (\boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{b})$?

- answer:

  define counts:

  $c_x = \sum_{i=1}^N [x_i = x]$

  $c_{yx} = \sum_{i=1}^N [y_{1,i} = y][x_i = x]$

  $c_{zx} = \sum_{i=1}^N [y_{2,i} = z][x_i = x]$

  set parameters equal:

  $\pi_x \leftarrow \frac{c_x}{N}$

  $a_{yx} \leftarrow \frac{c_{yx}}{N c_x}$

  $b_{zx} \leftarrow \frac{c_{zx}}{N c_x}$

Learning with complete data

- why (in what sense) is this the correct answer?

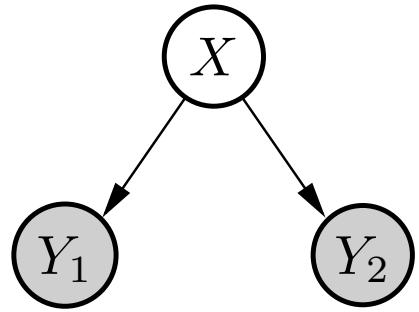
  – these parameters maximize *observed data log-likelihood*

$$L(\theta) = \log \prod_{i=1}^{n} P(x_i, \boldsymbol{y}_i \,; \theta)$$

$$= \sum_{i=1}^{n} \log P(x_i, \boldsymbol{y}_i \,; \theta)$$

Learning with missing data

- Given three random varibles X, Y_1, Y_2 with structure



$$P(X, Y_{1,2}) = P(X)\ P(Y_1|X)\ P(Y_2|X)$$

  – unknown parameters $P(X=x) \equiv \pi_x$, $P(Y=y|X=x) \equiv a_{yx}$, $P(Y_2=z|X=x) \equiv b_{zx}$

  – Given a partial data set $\{(\boldsymbol{y}_i)\}_{i=1}^N$ (observations x_i are missing!)

  – how can we learn the parameters θ?

- any ideas?

General ideas for learning with missing data

- We should somehow *fill in the missing data*..

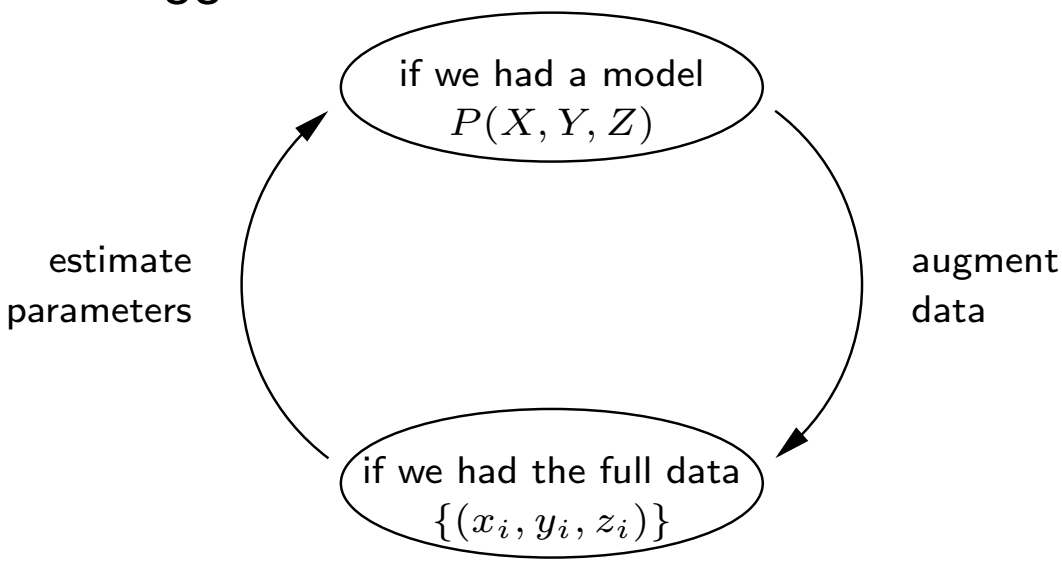
$$\text{partial data } \{(\boldsymbol{y}_i)\}_{i=1}^N \quad \rightarrow \quad \text{augmented data } \{(\hat{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$$

- but how should we choose $\hat{x}_i$
  ... invent fictional $\hat{x}_i$??

A chicken and egg problem

- If we knew the model $P(X, Y)$ already, we could use it to invent/estimate a $\hat{x}_i$ for each partial datum $\boldsymbol{y}_i$
  – then use this "augmented data" to train the model

- the chicken and egg situation:



Expectation Maximization

- instead of strict augmentation $\hat{x}_i = \text{argmax}_x \, P(x, \boldsymbol{y}_i \, ; \theta^{\text{old}})$
  we can compute the posterior belief over the missing data using the current model

$$q_i(x) = P(x|\boldsymbol{y}_i \, ; \theta^{\text{old}})$$

  $\rightarrow$ then we have an "expected data augmentation"

- how choose new parameters θ?
  – for complete data, we optimize the data log-likelihood
  – now, we can optimzie the *expected* data log-likelihood...

Expected data log-likelihood

- complete data log-likelihood (if X and Y are observed):

$$L(\theta) = \sum_{i=1}^{n} \log P(x_i, y_i \,; \theta)$$

- observed data log-likelihood (if only Y is observed and we can eliminate X analytically – often intractable):

$$\hat{L}(\theta) = \sum_{i=1}^{n} \log P(y_i \,; \theta) = \sum_{i=1}^{n} \log \sum_{x} P(x, y_i \,; \theta)$$

- expected data log-likelihood (if only Y is observed and we have a posterior $q_i(x; \theta^{\text{old}})$ over the missing data):

$$Q(\theta, \theta^{\text{old}}) = \sum_{i=1}^{n} \sum_{x} q_i(x; \theta^{\text{old}}) \; \log P(x, y_i \,; \theta)$$

$$\text{where} \;\; q_i(x; \theta^{\text{old}}) = P(x | y_i \,; \theta^{\text{old}})$$

Note: $Q(\theta, \theta^{\text{old}}) \leq \hat{L}(\theta)$ (see later, free energy)

Expectation Maximization II

- given partial data $\{y_i\}_{i=1}^N$
- given *some initial* parameters θ^{old} that define a model $P(X, Y\ ; \theta^{\text{old}})$
- iterate:

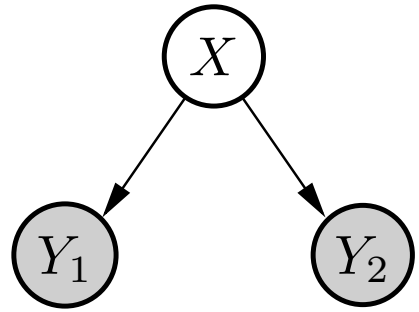
  (E-step) for each datum compute $q_i(x; \theta^{\text{old}}) = P(x|y_i\ ; \theta^{\text{old}})$
  ("expected data agumentation" using the old parameters)

  (M-step) compute new parameters

  $$\theta^{\text{new}} = \operatorname*{argmax}_{\theta} Q(\theta, \theta^{\text{old}})$$

  that maximize expected data log-likelihood

Example



$$P(X, Y, Z) = P(X)\ P(Y|X)\ P(Z|X)$$

- **(E-step)** Compute

$$q_i(x\ ; \theta^{\text{old}}) = P(x|y_i, z_i\ ; \theta^{\text{old}})$$
$$\propto P(x)\ P(y_i\,|\,x)\ P(z_i|x) = \pi_x^{\text{old}}\ a_{y_i x}^{\text{old}}\ b_{z_i x}^{\text{old}}$$

  This is an inference problem! – Naive Bayes!

- **(M-step)** compute *expected* counts:

  $c_x = \sum_{i=1}^{N} q_i(x)$
  $c_{yx} = \sum_{i=1}^{N} q_i(x)\ [y_i = y]$
  $c_{zx} = \sum_{i=1}^{N} q_i(x)\ [z_i = z]$

  and set parameter as before

- EM on an intuitive level:
  – *fill in missing data*
  – do this by computing the posterior $q_i(x\ ; \theta^{\text{old}})$ over missing variables
  – maximized expected data log-likelihood

- there exists very elegant and powerful theoretical formulation...

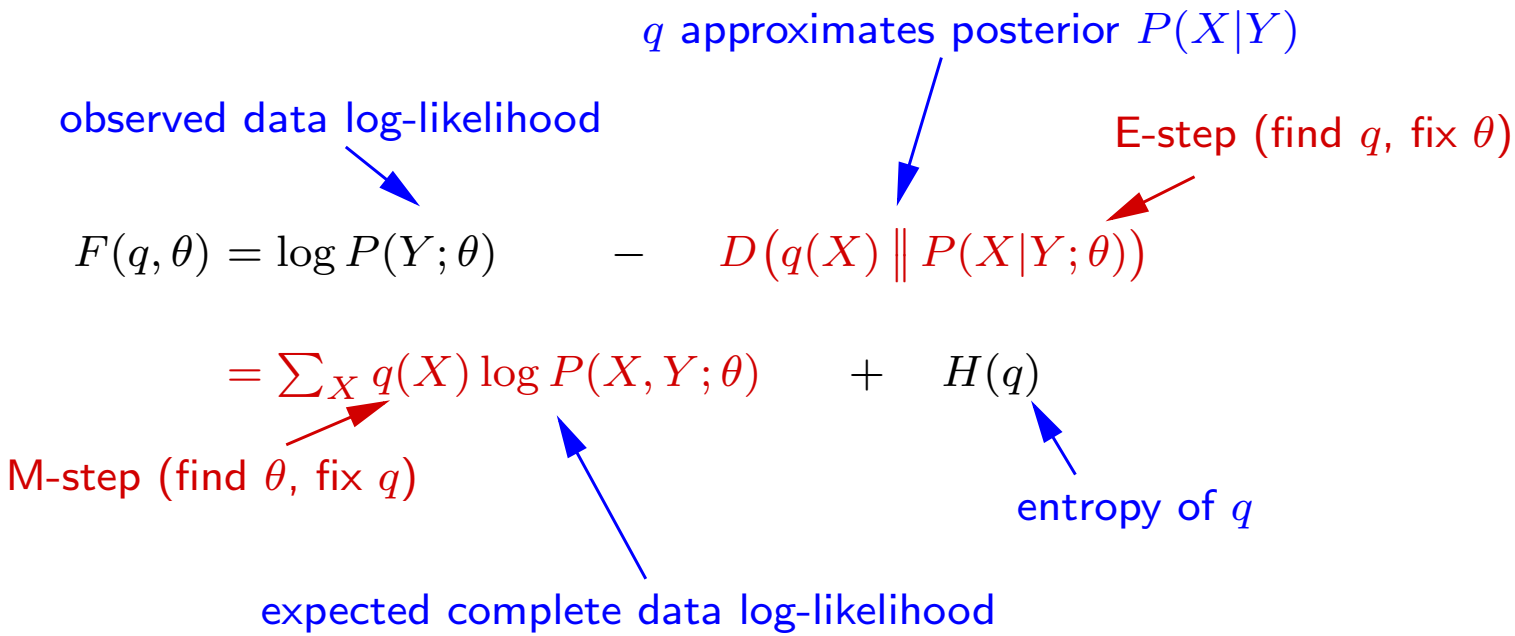
Free energy view on EM I

- Generally,
  - let X be a (set of i.i.d.) hidden variables
  - let Y be a (set of i.i.d.) observed variables
  - let $P(X, Y\ ; \theta)$ be a parameterized probabilistic model

- define the function

$$
\begin{aligned}
F(q, \theta) &= \log P(Y; \theta) - D\big(q(X) \,\big\|\, P(X|Y; \theta)\big) && (1)\\
&= \log P(Y; \theta) - \sum_X q(X) \log \frac{q(X)}{P(X \mid Y; \theta)} \\
&= \sum_X q(X) \log P(Y; \theta) + \sum_X q(X) \log P(X \mid Y; \theta) + H(q) \\
&= \sum_X q(X) \log P(X, Y; \theta) + H(q), && (2)
\end{aligned}
$$

Free energy view on EM II



- we actually want to maximize $P(Y;\theta)$ w.r.t. $\theta \rightarrow$ but can't analytically
- instead, maximize lower bound $F(q,\theta) \leq \log P(Y;\theta)$
  - E-step: find q that maximizes $F(q,\theta)$ for fixed θ^{old} using (1)
    ($\rightarrow$ find q to minimize KLD, makes lower bound tight for fixed θ)
  - M-step: find θ that maximizes $F(q,\theta)$ for fixed q using (2)
- EM = step-wise coordinate ascent of the function $F(q,\theta)$

$\Rightarrow$ convergence proof: F can only increase!

- that's it for learning in graphical models in this lecture

Summary 1

- probability distributions to express information/uncertainty
  (Bayesian vs. frequentist (description of repeatable experiments) view on probabilities)
  David MacKay: *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003

- Graphical models
  = describe joint probability in terms of factors
  $\rightarrow$ allow information processing between multiple variables

- inference as a general form of *Information Processing*
  – data is information
  – sensors give information
  – true state/actions/decisions are *missing* information (to be 'inferred')

- Message passing as powerful inference method

Summary 2

- Maximum Likelihood learning $L(\theta) = \sum_{i=1}^{n} \log P(x_i\ ; \theta)$
- Expectation Maximization: learning $P(X, Y)$ without observing X...
  - EM $\leftrightarrow$ idea of *fill in missing data*
  - EM $\leftrightarrow$ consider *expected* data log-likelihood
  - EM $\leftrightarrow$ free energy maximization as lower bound of observed data LL

Summary 3

we addressed

- *information processing*, in terms of probabilistic inference, message passing, multiplying, marginalizing, etc

- *learning*, in the sense of learning how information/RVs are coupled (also to input) $\leftrightarrow$ learning parameters of joint (or conditional) distributions

- further resources:

  https://ml01.zrz.tu-berlin.de/wiki/Main/SS09_GraphicalModels

thanks for your attention!