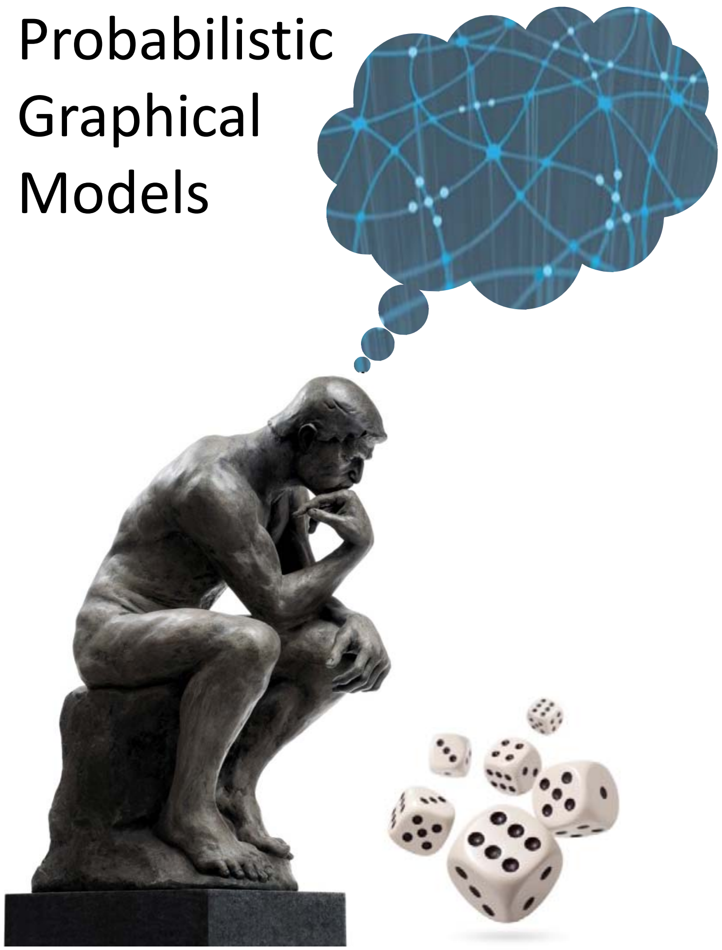
Probabilistic
Graphical
Models

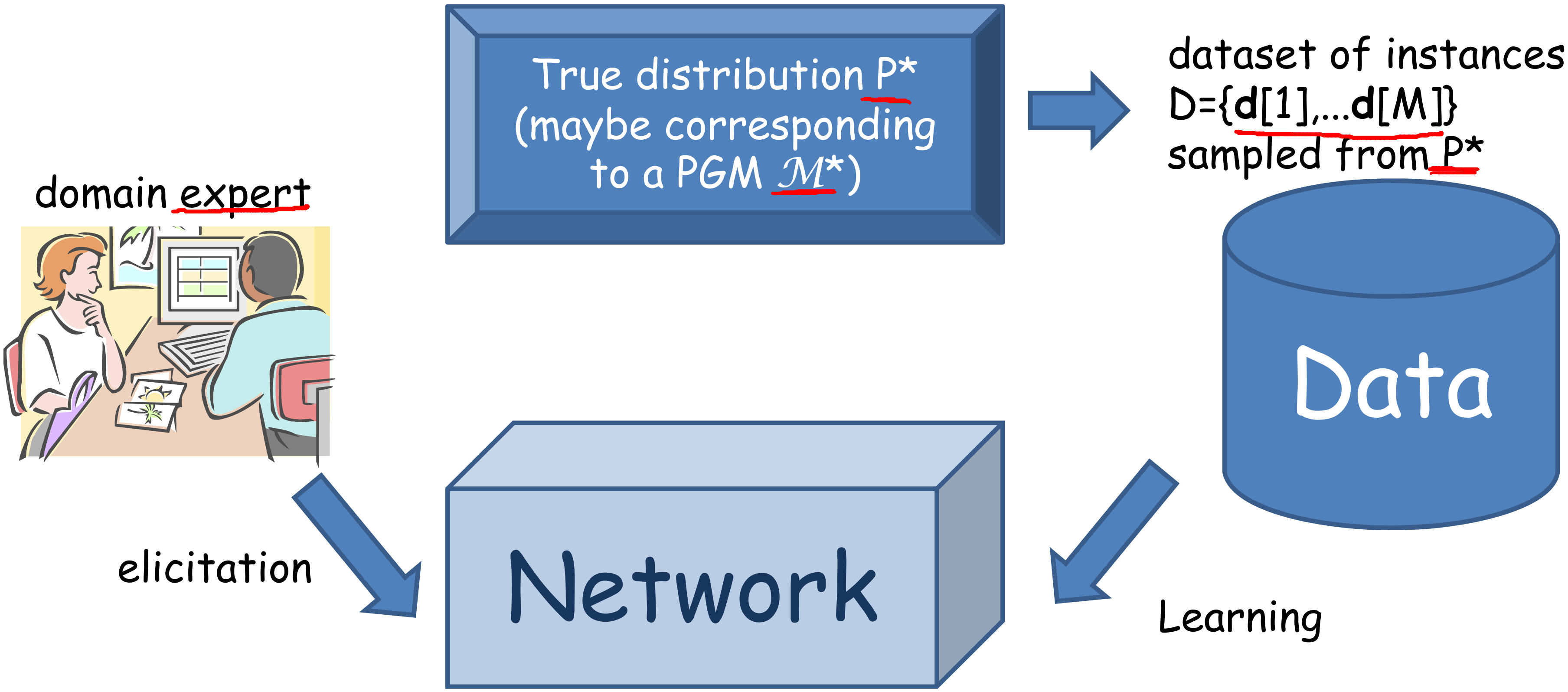


Learning

Overview

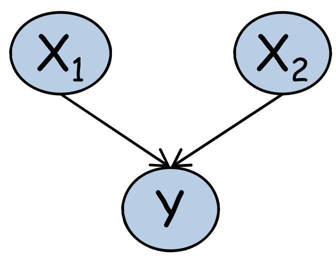
PGM Learning Tasks and Metrics

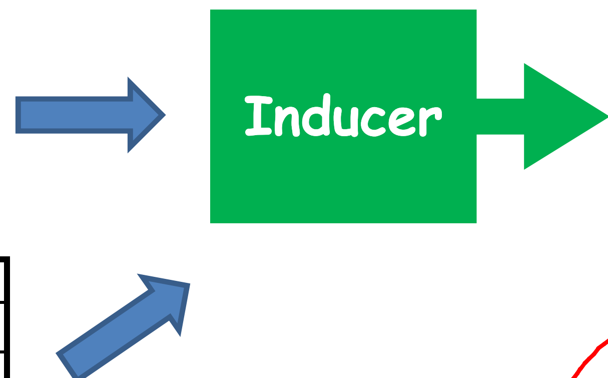
Learning

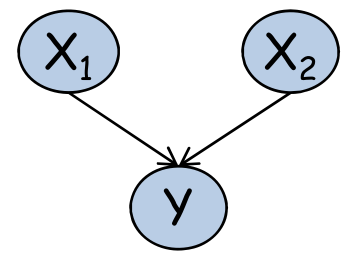

True distribution P*
(maybe corresponding to a PGM $\mathcal{M}$*)
dataset of instances
D={**d**[1],…**d**[M]}
sampled from P*
domain expert
Data
elicitation
Network
Learning

Known Structure, Complete Data

Initial network







Input Data

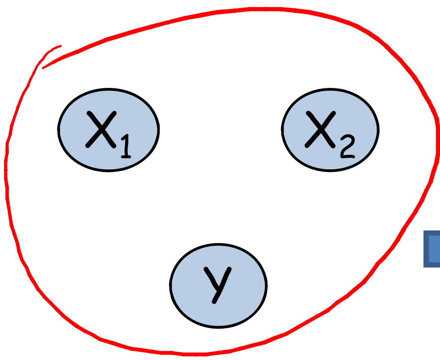
X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

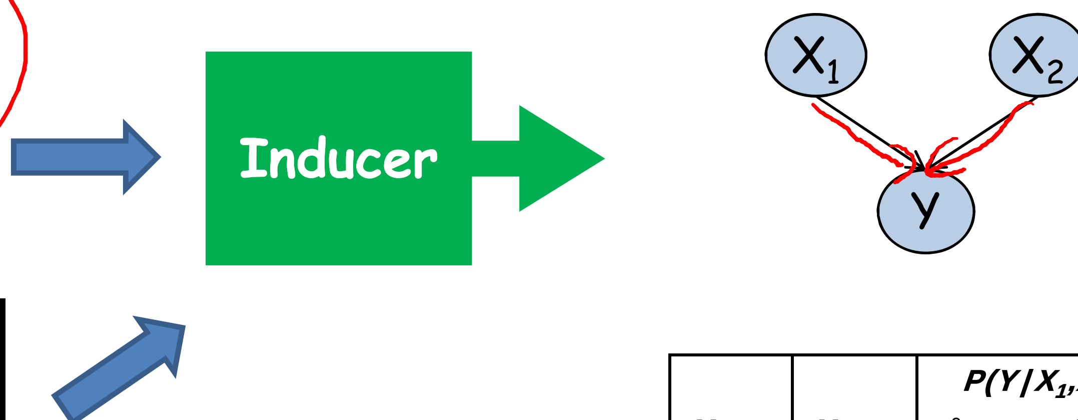
		$P(Y \mid X_1, X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Daphne Koller

Unknown Structure, Complete Data

Initial network





Input Data

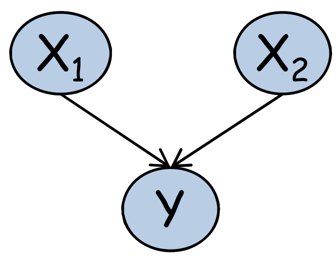
X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

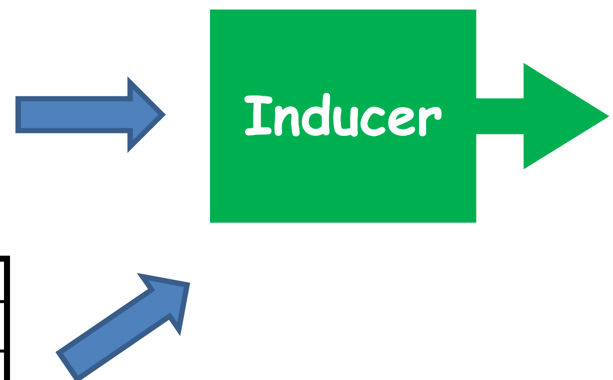
		$P(Y \mid X_1, X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

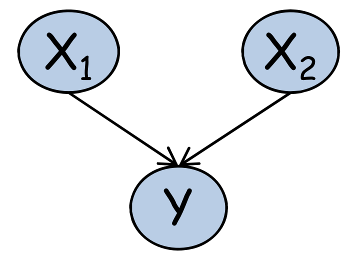
Daphne Koller

Known Structure, Incomplete Data

Initial network







Input Data

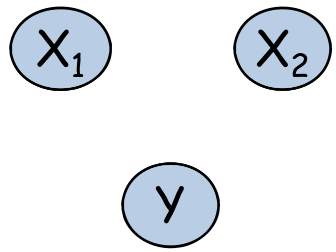
X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

		$P(Y \mid X_1,X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Daphne Koller

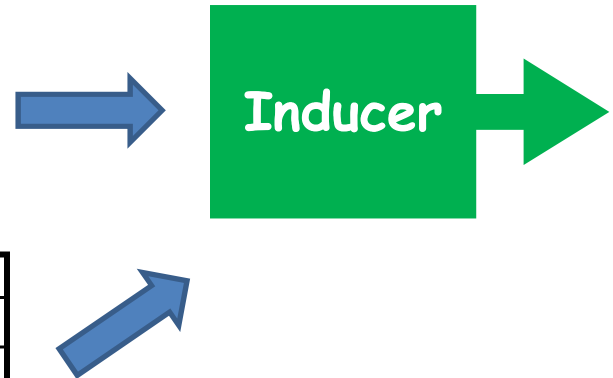
Unknown Structure, Incomplete Data

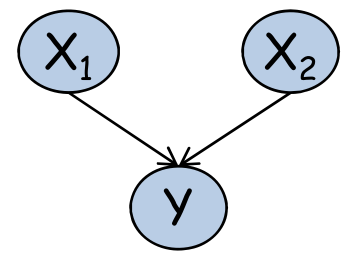
Initial network



Input Data

X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0



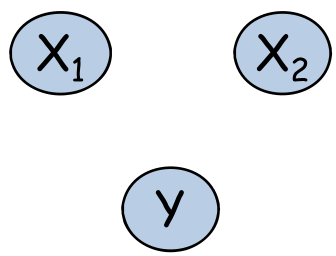


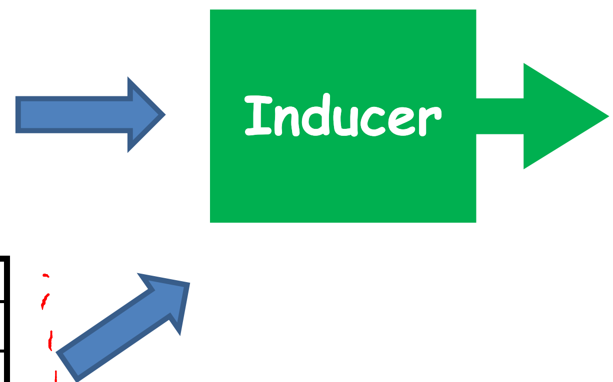
X_1	X_2	$P(Y \mid X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

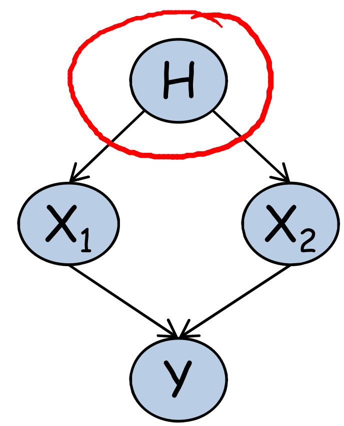
Daphne Koller

Latent Variables, Incomplete Data

Initial network







Input Data

X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

		$P(Y \mid X_1, X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Daphne Koller

PGM Learning Tasks I

- Goal: Answer general probabilistic queries about new instances
- Simple metric: Training set likelihood

  data

  – $P(D : \mathcal{M}) = \Pi_m P(\mathbf{d}[m] : \mathcal{M})$ (IID)
- But we really care about new data
  – Evaluate on test set likelihood – $P(D' : \mathcal{M})$

  generalization performance

PGM Learning Tasks II

- Goal: Specific prediction task on new instances
  - Predict target variables **y** from observed variables **x**
  - E.g., image segmentation, speech recognition
- Often care about specialized objective
  - E.g., pixel-level segmentation accuracy
- Often convenient to select model to optimize
  - likelihood $\Pi_m P(\mathbf{d}[m] : \mathcal{M})$ or
  - conditional likelihood $\Pi_m P(\mathbf{y}[m] \mid \mathbf{x}[m] : \mathcal{M})$
- Model evaluated on "true" objective over test data

Daphne Koller

PGM Learning Tasks III

- Goal: Knowledge discovery of $\mathcal{M}^*$
  - Distinguish direct vs indirect dependencies
  - Possibly directionality of edges
  - Presence and location of hidden variables
- Often train using likelihood
  - Poor surrogate for structural accuracy
- Evaluate by comparing to prior knowledge

Avoiding Overfitting

- Selecting $\mathcal{M}$ to optimize training set likelihood overfits to statistical noise
- Parameter overfitting
  - Parameters fit random noise in training data
  - Use regularization / parameter priors
- Structure overfitting
  - Training likelihood always increases for more complex structures
  - Bound or penalize model complexity

Selecting Hyperparameters

- Regularization for overfitting involves hyperparameters:
  - Parameter priors (regularization)
  - Complexity penalty
- Choice of hyperparameters makes a big difference to performance ~~training~~ ~~test~~
- Must be selected on validation set (cross-validation)

Why PGM Learning

- Predictions of structured objects (sequences, graphs, trees)
  - Exploit correlations between several predicted variables
- Can incorporate prior knowledge into model
- Learning single model for multiple tasks
- Framework for knowledge discovery

Probabilistic
Graphical
Models

Learning

Parameter Estimation

Maximum Likelihood Estimation

Biased Coin Example

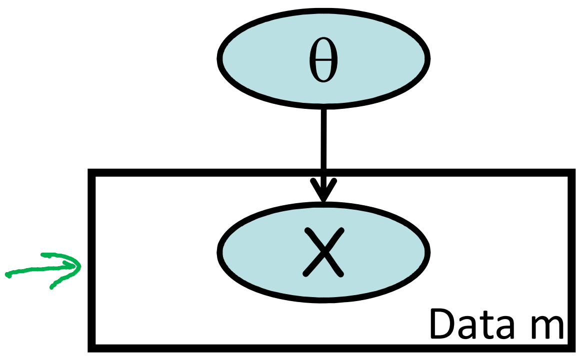
P is a Bernoulli distribution:

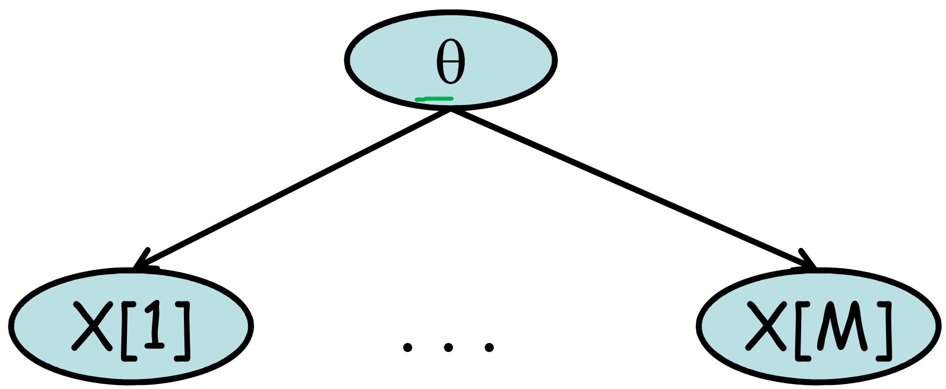
$P(X=1) = \theta$, $P(X=0) = 1-\theta$

$\mathcal{D} = \{x[1], \ldots, x[M]\}$ sampled IID from P

- Tosses are independent of each other
- Tosses are sampled from the same distribution (identically distributed)

IID as a PGM





$$P(x[m] \mid \theta) = \begin{cases} \theta & x[m] = x^1 \\ 1 - \theta & x[m] = x^0 \end{cases}$$

Daphne Koller

Maximum Likelihood Estimation

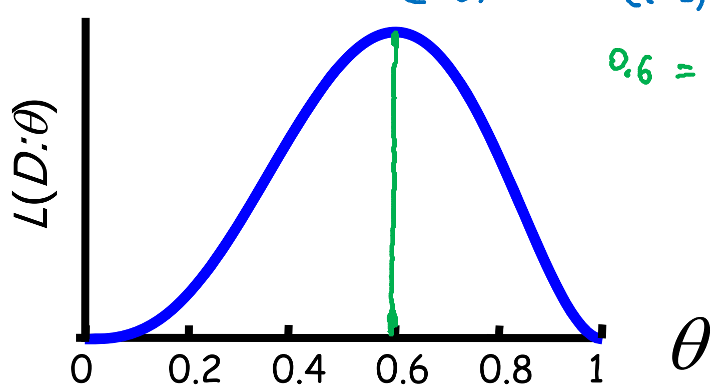
- Goal: find $\theta \in [0,1]$ that predicts D well
- Prediction quality = likelihood of D given θ

$$L(\theta : D) = P(D \mid \theta) = \prod_{m=1}^{M} P(x[m] \mid \theta)$$

$$L(\theta : \langle H, T, T, H, H \rangle)$$

$\underbrace{P(H \mid \theta)}_{\theta} \cdot \underbrace{P(T \mid \theta)}_{(1-\theta)} \cdot \underbrace{P(T \mid \theta)}_{(1-\theta)} \cdot \underbrace{P(H \mid \theta)}_{\theta} \cdot \underbrace{P(H \mid \theta)}_{\theta} = \theta^3 (1-\theta)^2$

$0.6 = \frac{3}{5}$



Daphne Koller

Maximum Likelihood Estimator

- Observations: M_H heads and M_T tails
- Find θ maximizing likelihood

log

$$L(\theta : M_H, M_T) = \theta^{M_H}(1-\theta)^{M_T}$$

- Equivalent to maximizing log-likelihood

$$l(\theta : M_H, M_T) = M_H \log\theta + M_T \log(1-\theta)$$

- Differentiating the log-likelihood and solving for θ: $\boxed{\hat{\theta} = \frac{M_H}{M_H + M_T}}$

Daphne Koller

Sufficient Statistics

- For computing θ in the coin toss example, we only needed M_H and M_T since

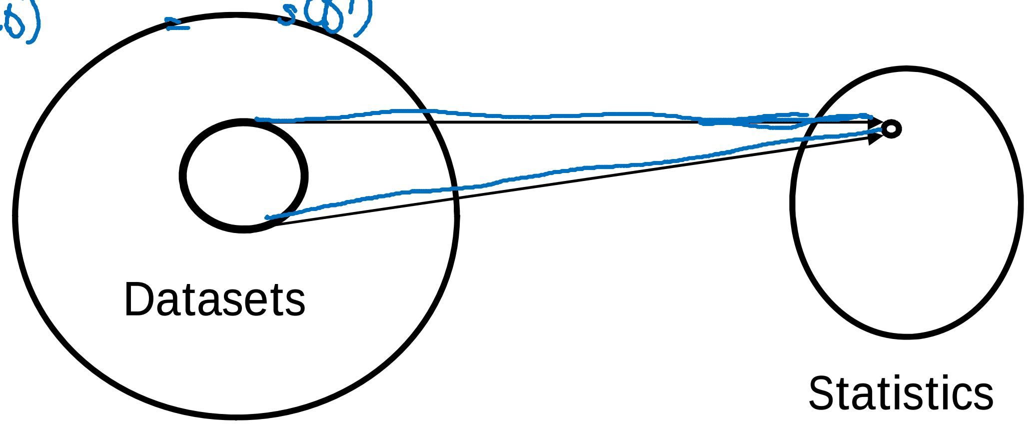
$$L(\theta : D) = \underline{\theta^{\boxed{M_H}}(1-\theta)^{\boxed{M_T}}}$$

- → M_H and M_T are sufficient statistics

Sufficient Statistics

- A function s(D) is a sufficient statistic from instances to a vector in $\Re^k$ if for any two datasets D and D′ and any $\theta \in \Theta$ we have

$$\underbrace{\sum_{x[i] \in D} s(x[i])}_{s(D)} = \underbrace{\sum_{x[i] \in D'} s(x[i])}_{s(D')} \quad \Rightarrow \quad L(\theta : D) = L(\theta : D')$$



Sufficient Statistic for Multinomial

- For a dataset D over variable X with k values, the sufficient statistics are counts $\langle M_1,\ldots,M_k \rangle$ where M_i is the # of times that $X[m]=x^i$ in D

- Sufficient statistic s(x) is a tuple of dimension k
  - $s(x^i)=(0,\ldots0,1,0,\ldots,0)$ $\quad \sum_m s(x[m]) = \{M_1, M_2, \ldots, M_k\}$

i

$$L(\theta : D) = \prod_{i=1}^{k} \theta_i^{M_i}$$

θ_i param for $X=x^i$

Daphne Koller

Sufficient Statistic for Gaussian

- Gaussian distribution:

$$P(X) \sim N(\mu, \sigma^2) \quad \textit{if} \quad p(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- Rewrite as

$$p(X) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-x^2 \frac{1}{2\sigma^2} + x\frac{\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2}\right)$$

- Sufficient statistics for Gaussian: s(x)=<1,x,x²>

$s(D) = \left(\sum_m x[m]^2, \sum_m x[m], M\right)$

Maximum Likelihood Estimation

- MLE Principle: Choose θ to maximize $L(D:\Theta)$

- Multinomial MLE:

$$\hat{\theta}^i = \frac{M_i}{\sum_{i=1}^m M_i}$$

fraction of x^i in data

- Gaussian MLE:

$$\hat{\mu} = \frac{1}{M}\sum_m x[m]$$

empirical mean

$$\hat{\sigma} = \sqrt{\frac{1}{M}\sum_m (x[m] - \hat{\mu})^2}$$

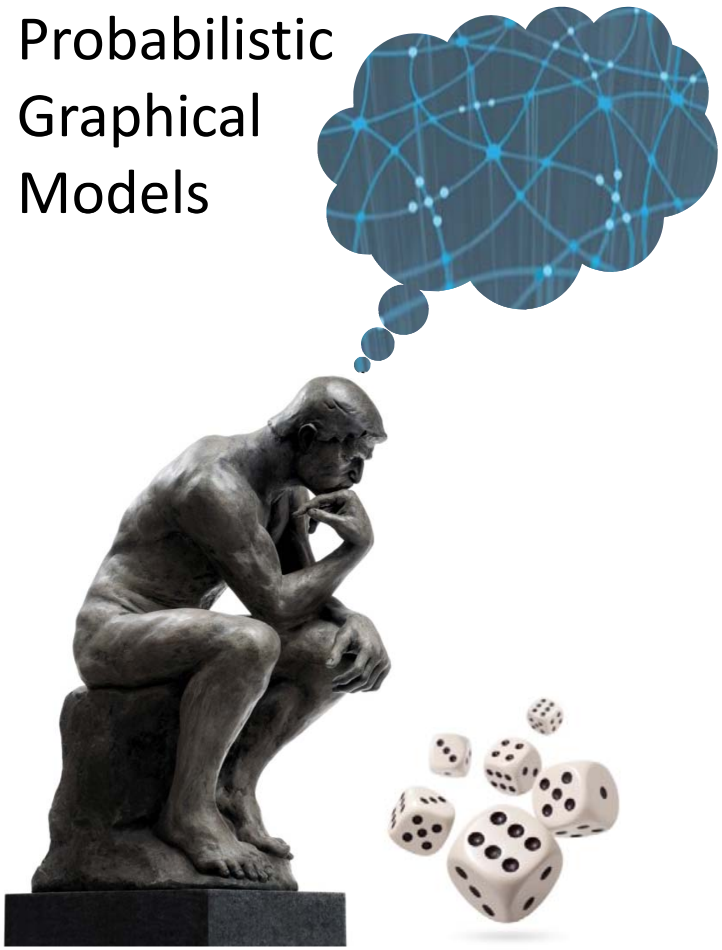
empirical st dev

Daphne Koller

Summary

- Maximum likelihood estimation is a simple principle for parameter selection given D
- Likelihood function uniquely determined by sufficient statistics that summarize D
- MLE has closed form solution for many parametric distributions

Probabilistic
Graphical
Models



Learning

Parameter Estimation

Max-Likelihood for BNs

MLE for Bayesian Networks

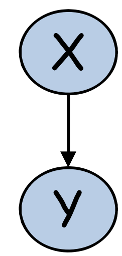
- Parameters: $\theta_{x^0}, \theta_{x^1}$

  $\theta_{y^0|x^0}, \theta_{y^1|x^0}, \theta_{y^0|x^1}, \theta_{y^1|x^1}$

- Data instances: <x[m],y[m]>

X	
x^0	x^1
0.7	0.3

P(X)



X	***Y*** y^0	y^1
x^0	0.95	0.05
x^1	0.2	0.8

P(Y|X)

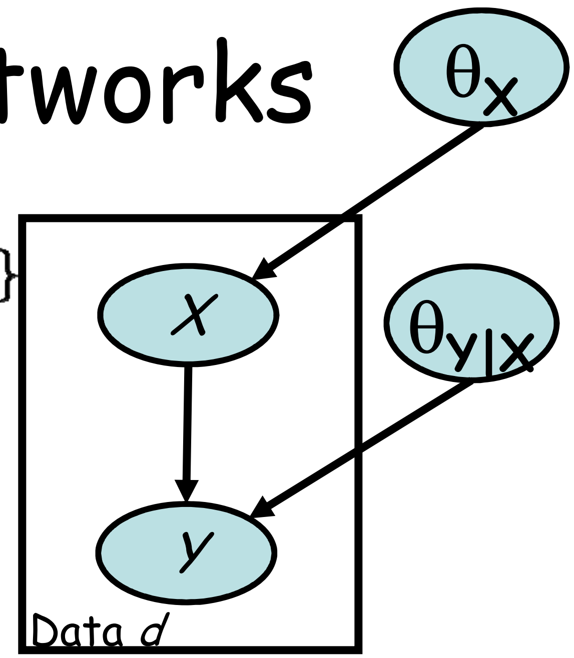
Daphne Koller

MLE for Bayesian Networks

- Parameters:

$$\{\theta_x : x \in Val(X)\}$$

$$\{\theta_{y|x} : x \in Val(X), y \in Val(Y)\}$$



$$L(\Theta : D) = \prod_{m=1}^{M} P(x[m], y[m] : \theta)$$

|| chain rule for BNs

$$= \prod_{m=1}^{M} P(x[m] : \theta) P(y[m] \mid x[m] : \theta)$$

$$= \left(\prod_{m=1}^{M} P(x[m] : \theta)\right)\left(\prod_{m=1}^{M} P(y[m] \mid x[m] : \theta)\right)$$

$$= \left(\prod_{m=1}^{M} P(x[m] : \theta_X)\right)\left(\prod_{m=1}^{M} P(y[m] \mid x[m] : \theta_{Y|X})\right)$$

product of two local likelihood

MLE for Bayesian Networks

- Likelihood for Bayesian network

parents of X_i

chain rule

$$
\begin{aligned}
L(\Theta : D) &= \prod_m P(x[m] : \Theta) \\
&= \prod_m \prod_i P(x_i[m] \mid \mathrm{U}_i[m] : \Theta_i) \\
&= \prod_i \underbrace{\prod_m P(x_i[m] \mid \mathrm{U}_i[m] : \Theta_i)}_{L_i(\theta_{X_i|U_i} : D)} \\
&= \prod_i L_i(\Theta_i : D)
\end{aligned}
$$

local likelihood

⇨ if $\theta_{X_i|U_i}$ are disjoint, then MLE can be computed by maximizing each local likelihood separately

MLE for Table CPDs

$$\prod_{m=1}^{M} P(x[m] \mid \mathbf{u}[m] : \theta) = \prod_{m=1}^{M} P(x[m] \mid \mathbf{u}[m] : \theta_{X|\mathbf{U}})$$

$$= \prod_{x,\mathbf{u}} \left(\prod_{m : x[m] = x, \mathbf{u}[m] = \mathbf{u}} P(x[m] \mid \mathbf{u}[m] : \theta_{X|\mathbf{U}}) \right)$$

$P(X[m]=x \mid u[m]=u : \theta_{x|u}) = \theta_{x|u}$

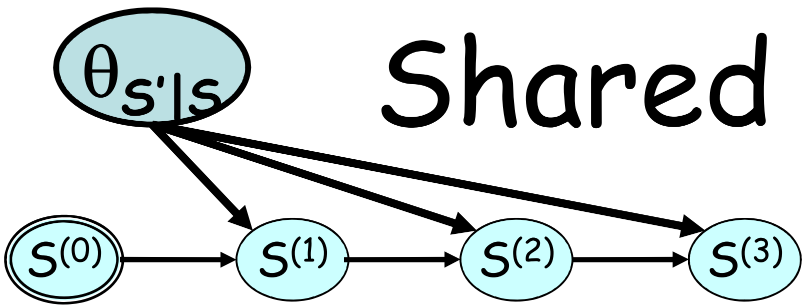
$$= \prod_{x,\mathbf{u}} \left(\prod_{m : x[m] = x, \mathbf{u}[m] = \mathbf{u}} \theta_{x|\mathbf{u}} \right)$$

$$= \prod_{x,\mathbf{u}} \underbrace{\theta_{x|\mathbf{u}}}_{P(x|u)}{}^{M[x,\mathbf{u}]}$$

fraction of $X=x$ among cases where $\bar{U}=\bar{u}$

$$\theta_{x|\mathbf{u}} = \frac{M[x,\mathbf{u}]}{\sum_{x'} M[x',\mathbf{u}]} = \frac{M[x,\mathbf{u}]}{M[\mathbf{u}]}$$

Shared Parameters



$$L(\theta : S^{(0:T)}) = \prod_{t=1}^{T} P(S^{(t)} \mid S^{(t-1)} : \theta)$$

$$= \prod_{i,j} \prod_{\underset{s^i \to s^j}{t:S^{(t)}=s^i, S^{(t+1)}=s^j}} P(S^{(t+1)} \mid S^{(t)} : \theta_{S'|S})$$

$$= \prod_{i,j} \prod_{t:S^{(t)}=s^i, S^{(t+1)}=s^j} \theta_{s^i \to s^j}$$

$$= \prod_{i,j} \theta_{s^i \to s^j}^{M[s^i \to s^j]}$$

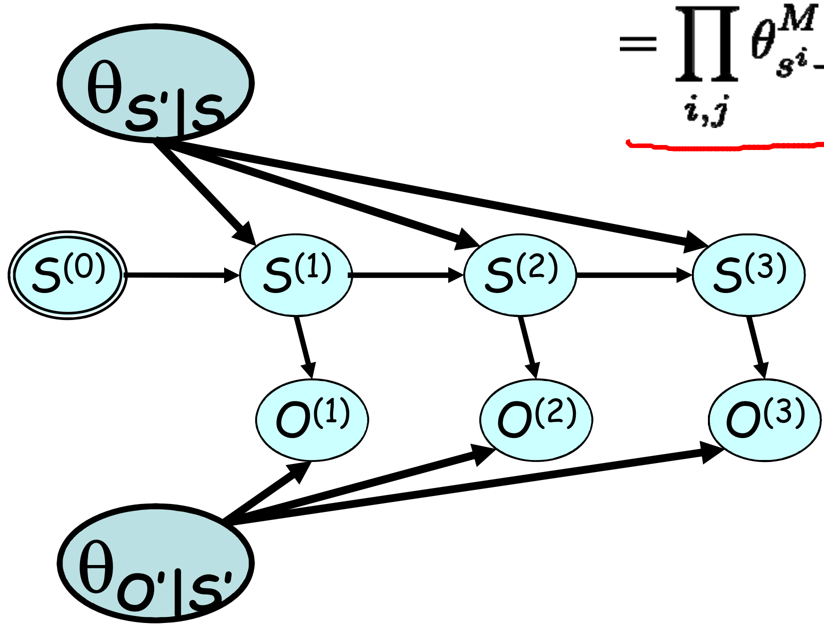
$$\hat{\theta}_{s^i \to s^j} = \frac{M[s^i \to s^j]}{M[s^i]}$$

$$M[s^i \to s^j] = |\{t \;:\; S^{(t)} = s^i, S^{(t+1)} = s^j\}|$$

Shared Parameters

$$L(\Theta : S^{(0:T)}, O^{(0:T)}) = \underbrace{\prod_{t=1}^{T} P(S^{(t)} \mid S^{(t-1)} : \theta_{S'|S})}\ \underbrace{\prod_{t=1}^{T} P(O^{(t)} \mid S^{(t)} : \theta_{O'|S'})}$$

$$= \underbrace{\prod_{i,j} \theta_{s^i \to s^j}^{M[s^i \to s^j]}}\ \prod_{i,k} \theta_{o^k|s^i}^{M[o^k, s^i]}$$



$$M[s^i \to s^j] = |\{t \ : \ S^{(t)} = s^i, S^{(t+1)} = s^j\}|$$

$$\underline{M[o^k, s^i]} = |\{t \ : \ S^{(t)} = \underline{s^i}, O^{(t)} = \underline{o^k}\}|$$

Daphne Koller

Summary

- For BN with disjoint sets of parameters in CPDs, likelihood decomposes as product of local likelihood functions, one per variable
- For table CPDs, local likelihood further decomposes as product of likelihood for multinomials, one for each parent combination
- For networks with shared CPDs, sufficient statistics accumulate over all uses of CPD

Fragmentation & Overfitting

$$\theta_{x|\mathbf{u}} = \frac{M[x,\mathbf{u}]}{\sum_{x'} M[x',\mathbf{u}]} = \frac{M[x,\mathbf{u}]}{M[\mathbf{u}]}$$

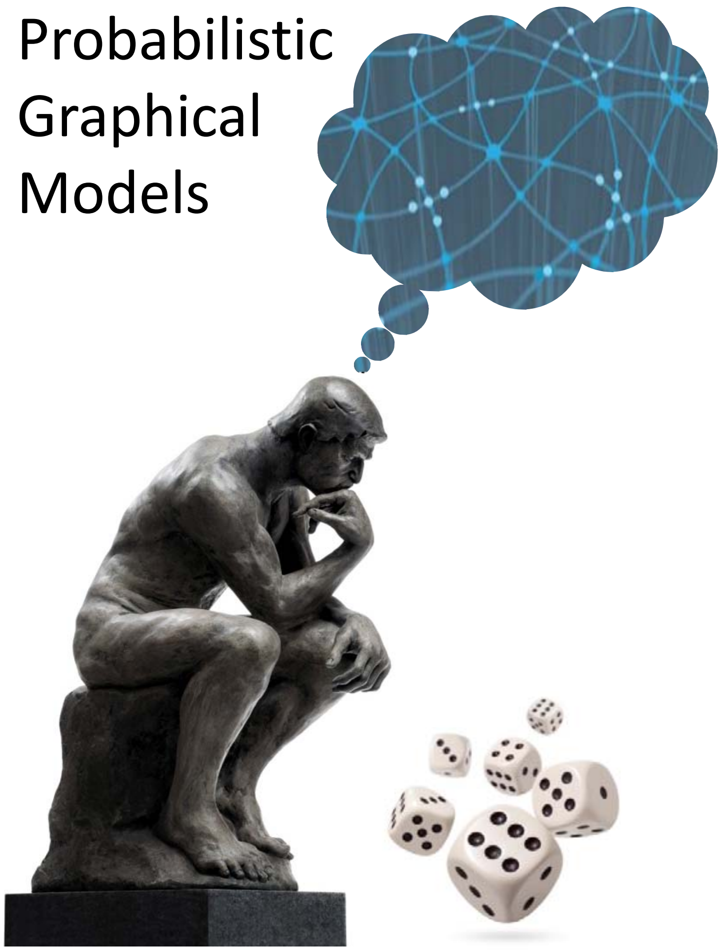
- # of "buckets" increases exponentially with |U|

u

- For large |U|, most "buckets" will have very few instances
  - ⇨ **very poor parameter estimates** ⇐
- With limited data, we often get better generalization with simpler structures

even when wrong

Probabilistic
Graphical
Models



Learning

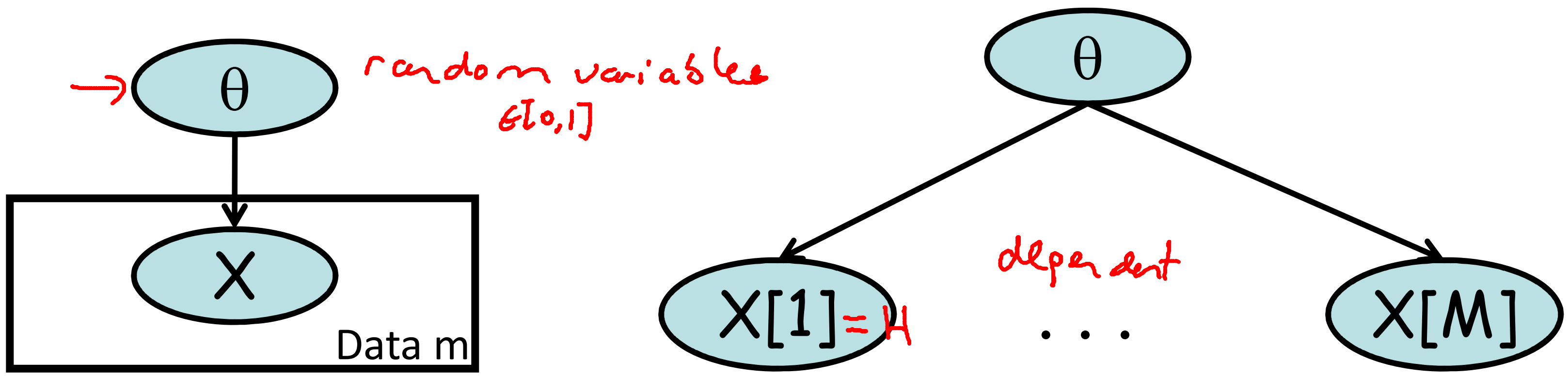
Parameter Estimation

Bayesian Estimation

Limitations of MLE

- Two teams play 10 times, and the first wins 7 of the 10 matches
  - ⇨ Probability of first team winning = 0.7
- A coin is tossed 10 times, and comes out 'heads' 7 of the 10 tosses
  - ⇨ Probability of heads = 0.7
- A coin is tossed 10000 times, and comes out 'heads' 7000 of the 10000 tosses
  - ⇨ Probability of heads = 0.7

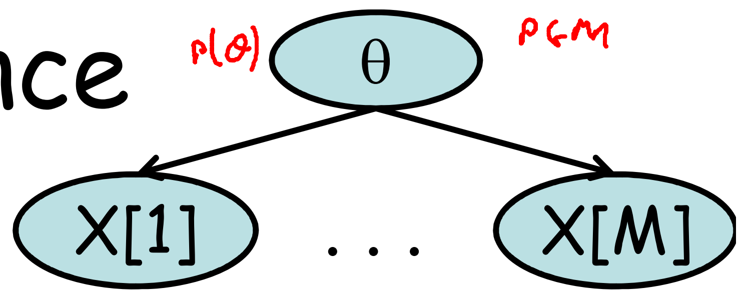
Parameter Estimation as a PGM



- Given a fixed θ, tosses are independent
- If θ is unknown, tosses are not marginally independent
  - each toss tells us something about θ

Bayesian Inference

- Joint probabilistic model



$$P(x[1],\ldots,x[M],\theta) = P(x[1],\ldots,x[M]\mid\theta)\,P(\theta)$$

$$= P(\theta)\prod_{i=1}^{M} P(x[i]\mid\theta)$$

likelihood function

$$= P(\theta)\,\theta^{M_H}(1-\theta)^{M_T}$$

$$P(\theta \mid \underbrace{x[1],\ldots,x[M]}_{\text{data } D}) = \frac{\overbrace{P(x[1],\ldots,x[M]\mid\theta)}^{\text{likelihood}}\,\overbrace{P(\theta)}^{\text{prior}}}{P(x[1],\ldots,x[M])}$$

constant relative to θ

Dirichlet Distribution

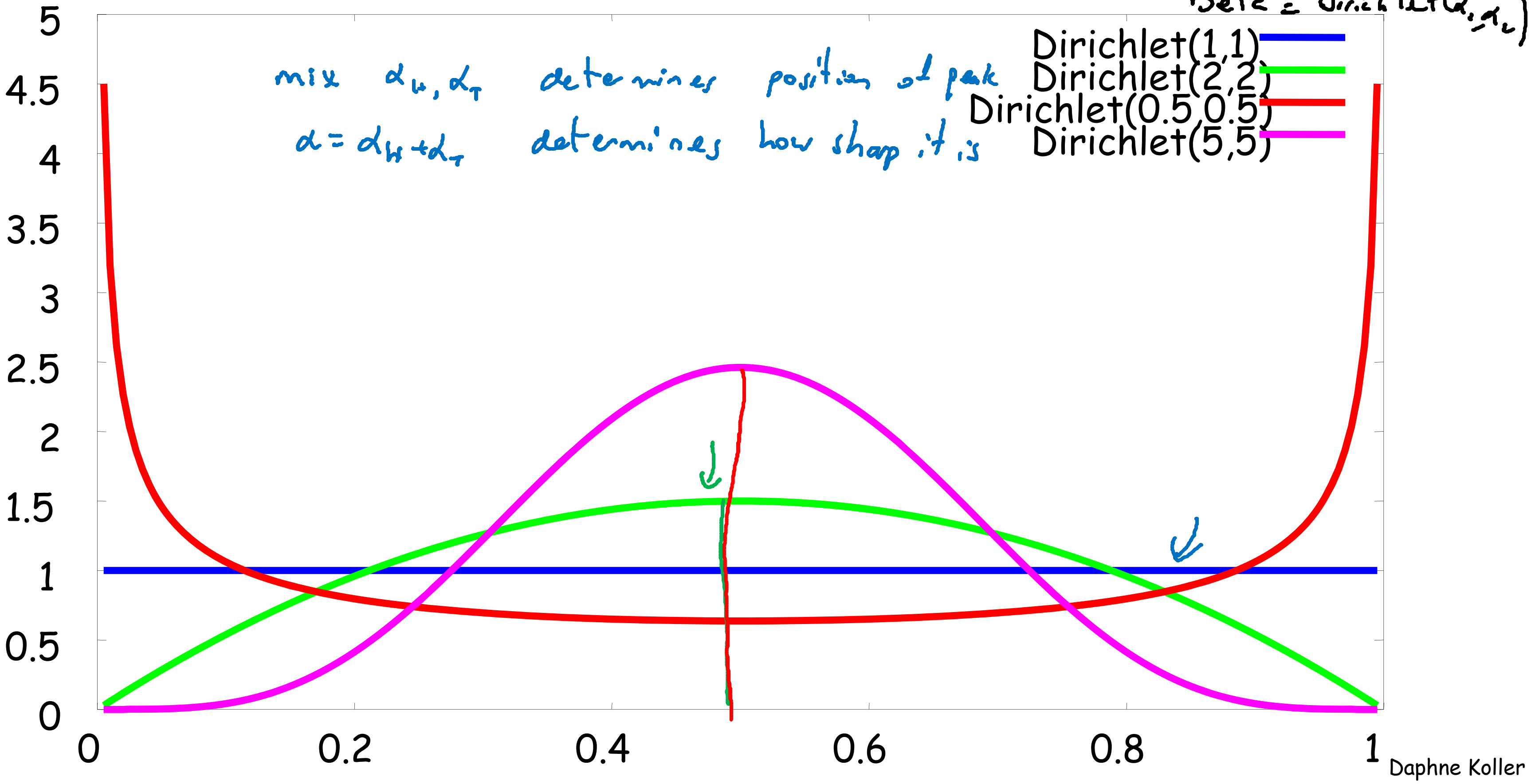
- θ is a multinomial distribution over k values
- Dirichlet distribution $\theta \sim \text{Dirichlet}(\underbrace{\alpha_1,\ldots,\alpha_k}_{\text{hyperparameters}})$
  - where $P(\theta) = \boxed{\frac{1}{Z}} \boxed{\prod_{i=1}^{k} \theta_i^{\alpha_i - 1}}$ and $Z = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}$ $\quad \Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t}\, dt$

α_i

- Intuitively, hyperparameters correspond to the number of samples we have seen

Dirichlet Distributions

Beta = Dirichlet(α_1, α_2)



Dirichlet Priors & Posteriors

$$\overbrace{P(\theta \mid D)}^{\text{posterior}} \propto \overbrace{P(D \mid \theta)}^{\text{likelihood}}\overbrace{P(\theta)}^{\text{prior}}$$

$\theta_i^{M_i+\alpha_i-1}$

multinomial θ

M_i = # instances with x^i

$$P(D \mid \theta) = \prod_{i=1}^{k} \theta_i^{M_i} \qquad P(\theta) \propto \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

- If $P(\theta)$ is Dirichlet and the likelihood is multinomial, then the posterior is also Dirichlet
  - Prior is $Dir(\alpha_1,\ldots,\alpha_k)$
  - Data counts are $M_1,\ldots,M_k$
  - Posterior is $Dir(\alpha_1+M_1,\ldots\alpha_k+M_k)$
- Dirichlet is a conjugate prior for the multinomial

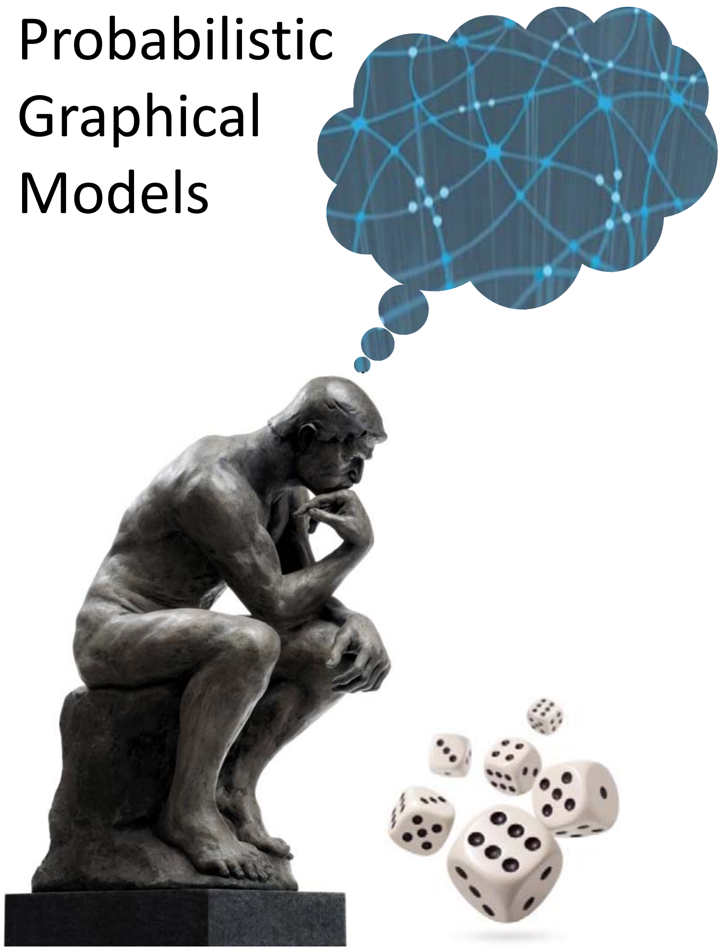
prior, posterior have the same form

Daphne Koller

Summary

- Bayesian learning treats parameters as random variables
  - Learning is then a special case of inference
- Dirichlet distribution is conjugate to multinomial
  - Posterior has same form as prior
  - Can be updated in closed form using sufficient statistics from data

Probabilistic
Graphical
Models

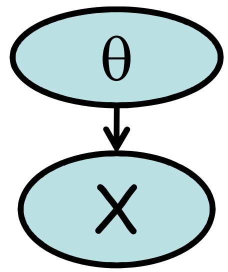


Learning

Parameter Estimation

Bayesian Prediction

Bayesian Prediction



~ Dirichlet($\alpha_1,\ldots,\alpha_k$)

marginalizing over θ

$$P(X) = \int_\theta P(X \mid \theta) P(\theta)\, d\theta$$

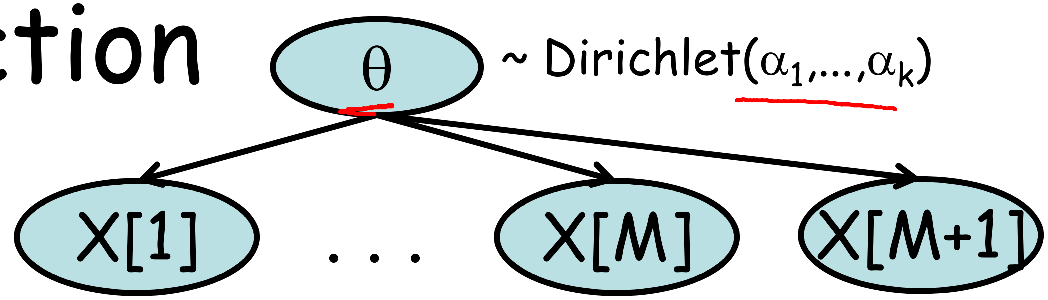
$$P(X = x^i \mid \theta) = \frac{1}{Z}\int_\theta \theta_i \cdot \underbrace{\prod_j \theta^{\alpha_j - 1}}_{\text{prior}} d\theta$$

$$= \frac{\alpha_i}{\sum_j \alpha_j = \alpha}$$

fraction of instances we've seen where x^i

- Dirichlet hyperparameters correspond to the number of samples we have seen

Bayesian Prediction



$$P(x[M+1] \mid x[1],\ldots,x[M])$$

$$= \int_\theta P(x[M+1] \mid x[1],\ldots,x[M],\theta)\, P(\theta \mid x[1],\ldots,x[M])\, d\theta$$

$$= \int_\theta P(x[M+1] \mid \theta)\, \boxed{P(\theta \mid x[1],\ldots,x[M])}\, d\theta$$

~Dirichlet($\alpha_1+M_1,\ldots,\alpha_k+M_k$)

posterior over θ given D

$$P(X[M+1] = x^i \mid \theta, x[1],\ldots,x[M]) = \frac{\alpha_i + M_i}{\alpha + M}$$

$\alpha = \sum \alpha_i$

$M = \sum M_i$

- Equivalent sample size $\alpha = \alpha_1 + \ldots + \alpha_K$
  - Larger α ⇨ more confidence in our prior

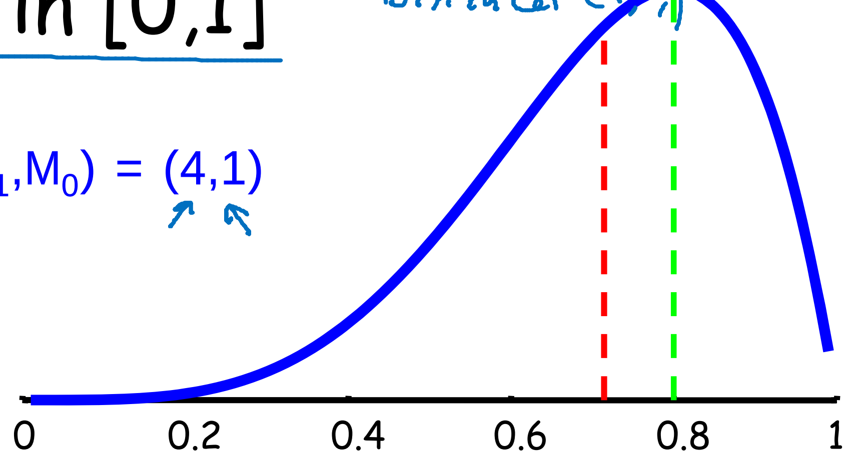
Example: Binomial Data

- Prior: uniform for θ in [0,1]

Dirichlet(1,1)

$$P(\theta) = \frac{1}{Z}\prod_k \theta_k^{\alpha_k - 1}$$

$(M_1, M_0) = (4,1)$



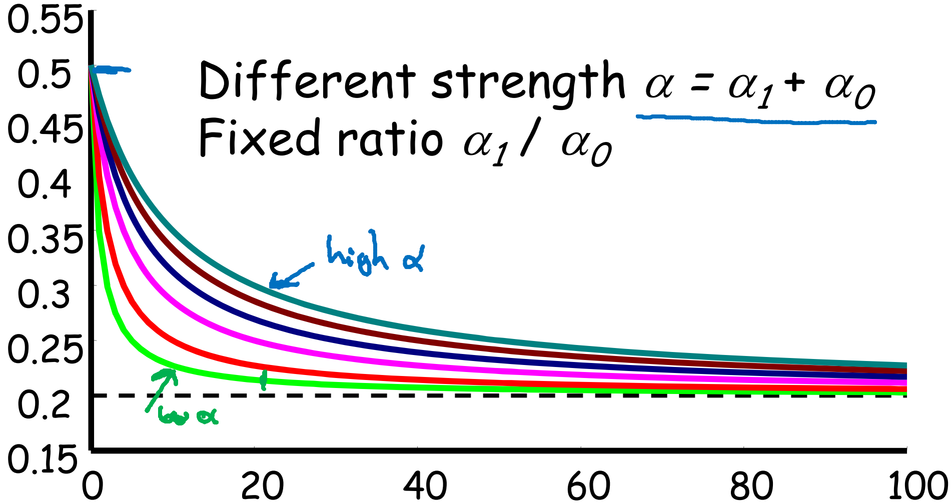
- MLE for P(X[6]=1)=4/5
- Bayesian prediction is 5/7

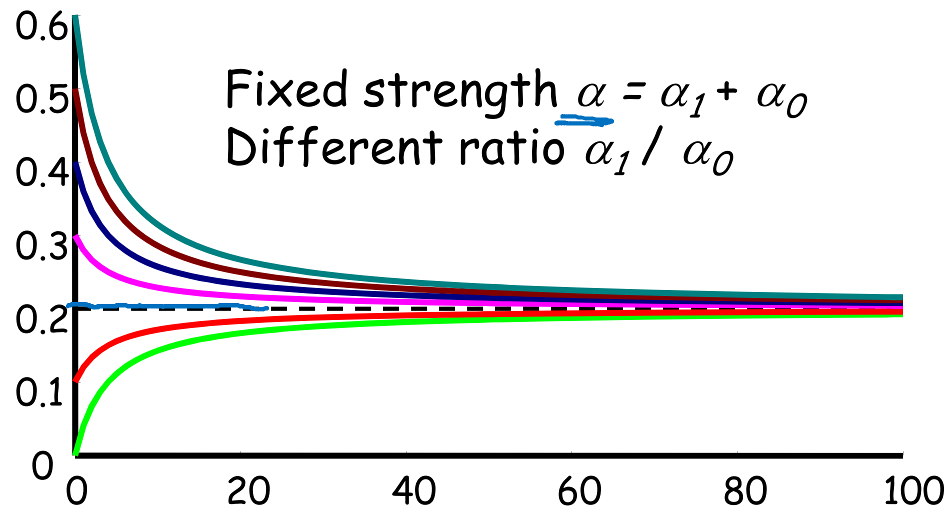
$$\frac{\alpha_1 + M_1}{\alpha + M} = \frac{1+4}{2+5}$$

Daphne Koller

Effect of Priors

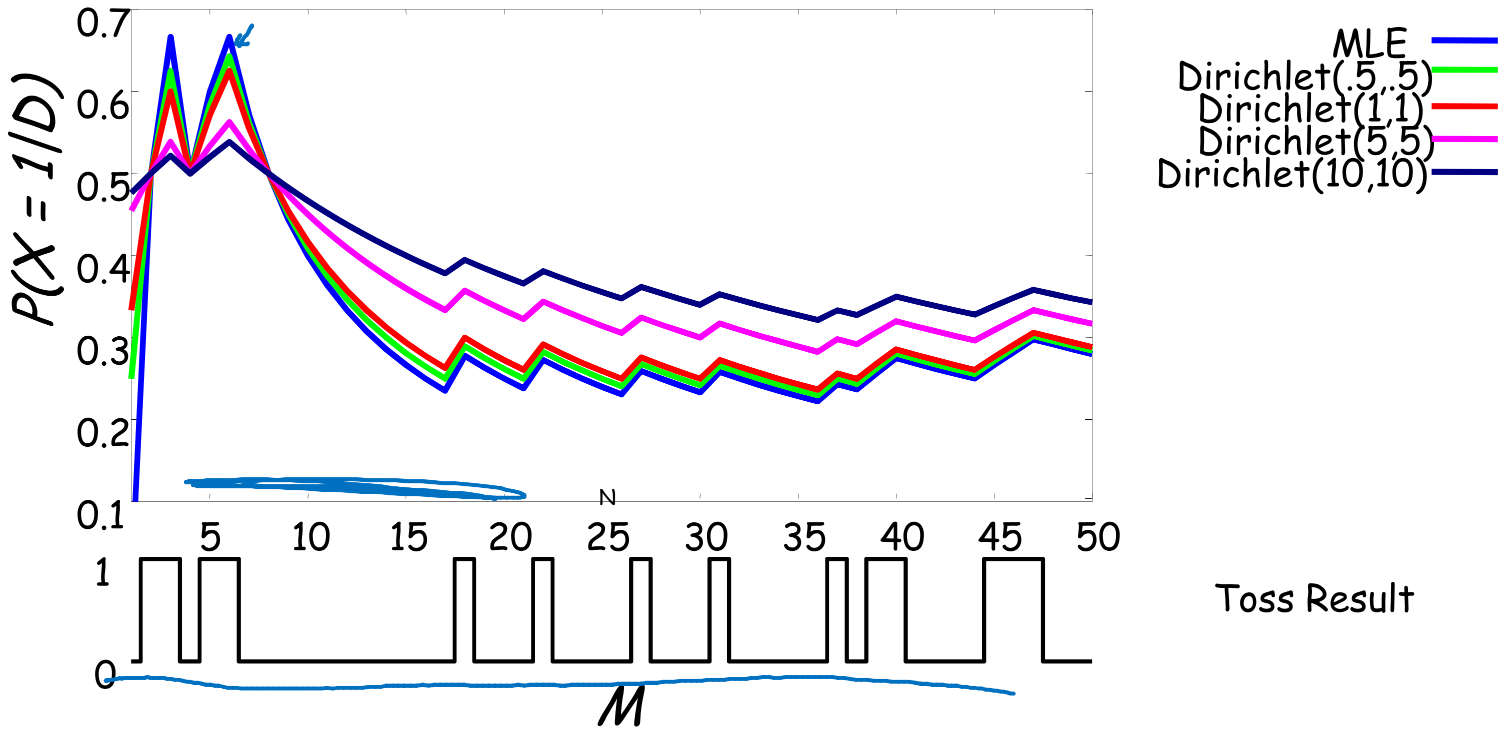
- Prediction of P(X=1) after seeing data with $M_1 = \frac{1}{4} M_0$ as a function of sample size M





Effect of Priors

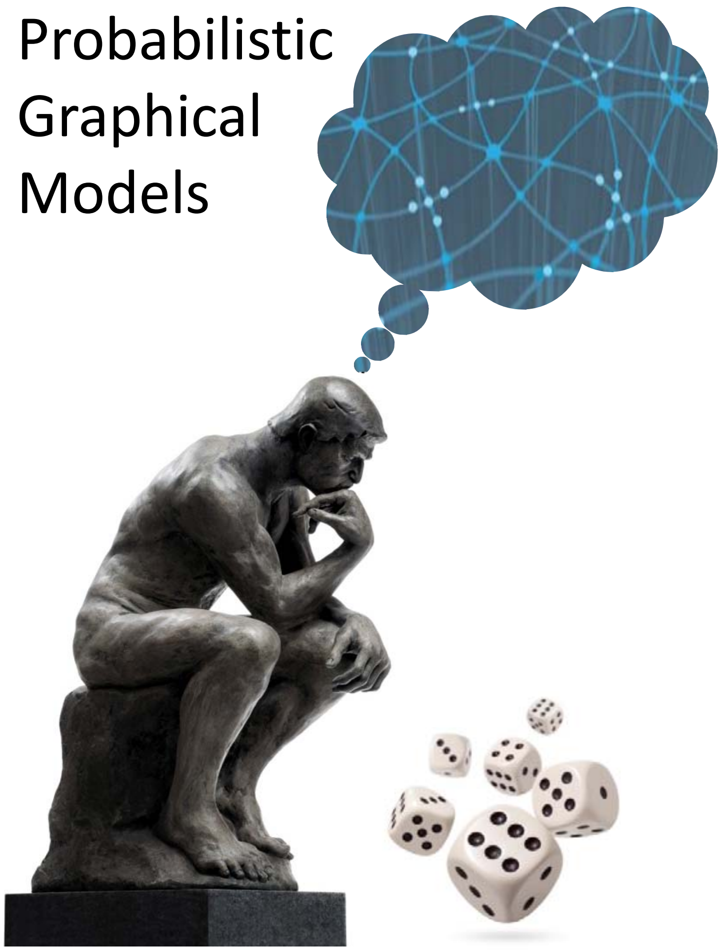
- In real data, Bayesian estimates are less sensitive to noise in the data



Summary

- Bayesian prediction combines sufficient statistics from imaginary Dirichlet samples and real data samples
- Asymptotically the same as MLE
- But Dirichlet hyperparameters determine both the prior beliefs and their strength

Probabilistic
Graphical
Models

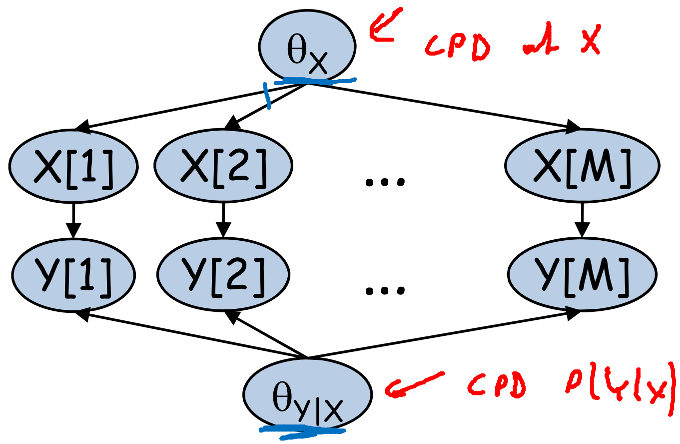


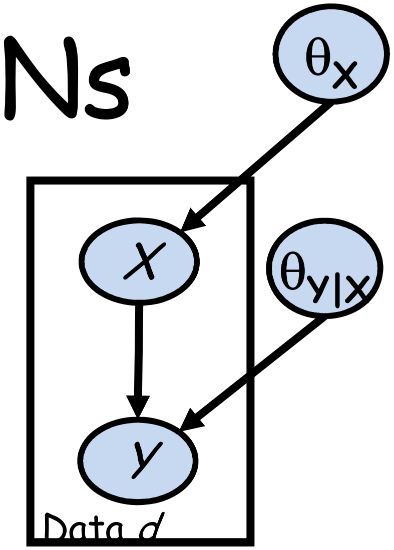
Learning

Parameter Estimation

Bayesian Estimation for BNs

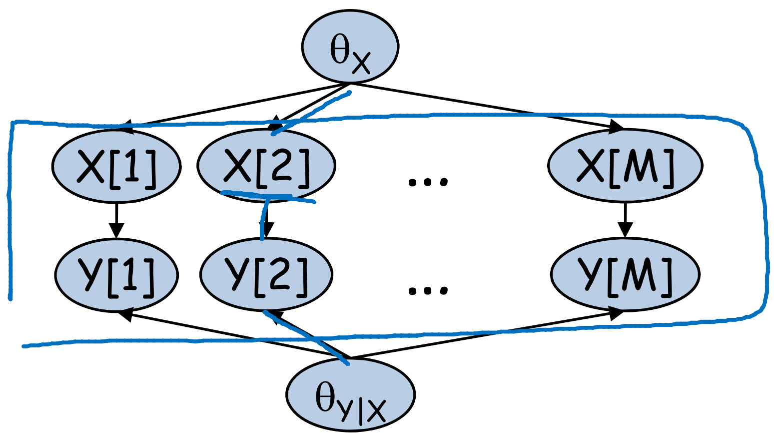
Bayesian Estimation in BNs

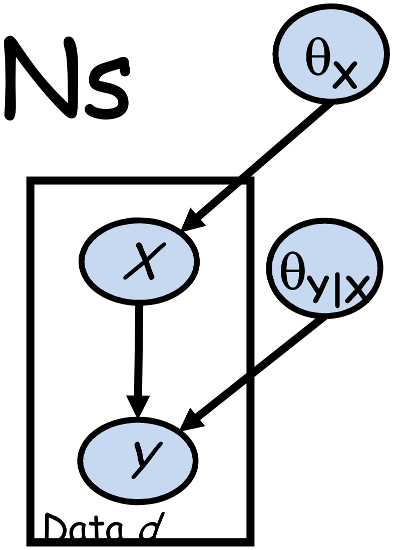




- Instances are independent given the parameters
  - (X[m'],Y[m']) are d-separated from (X[m],Y[m]) given θ
- Parameters for individual variables are independent a priori $P(\theta) = \prod_i P(\theta_{X_i|Pa(X_i)})$

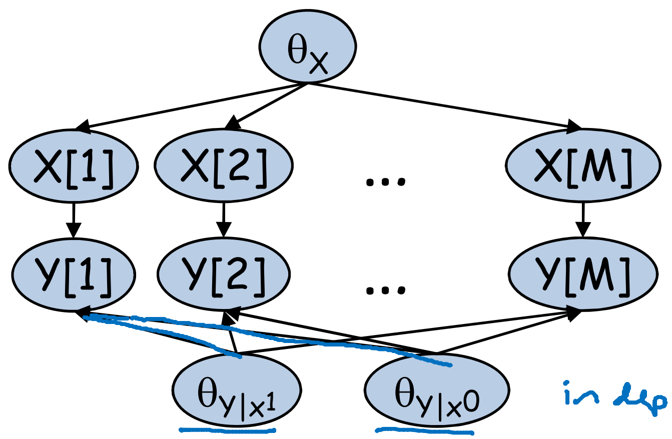
Bayesian Estimation in BNs

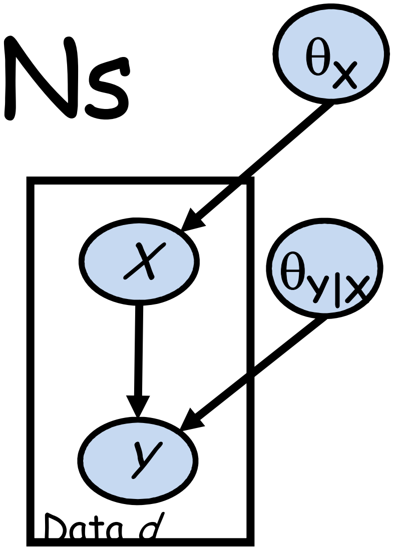




- Posteriors of θ are independent given complete data
  - Complete data d-separates parameters for different CPDs
  - $P(\theta_X, \theta_{Y|X} \mid D) = P(\theta_X \mid D) P(\theta_{Y|X} \mid D)$
  - As in MLE, we can solve each estimation problem separately

Bayesian Estimation in BNs





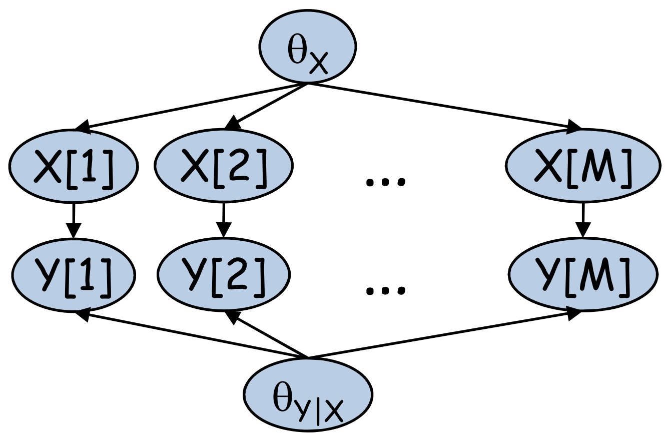
$P(\theta_X | D) \cdot P(\theta_{Y|x} | D) \cdot P(\theta_{Y|x} | D)$

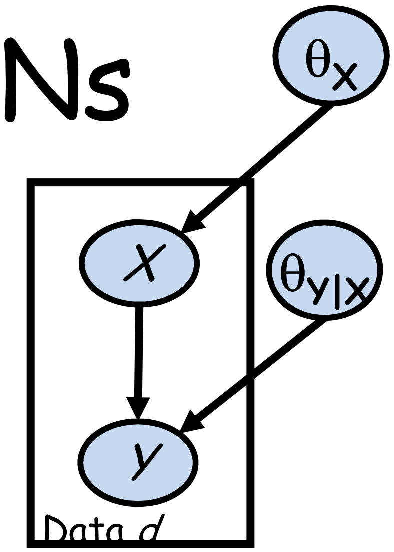
independent a priori

- Posteriors of θ are independent given complete data
  - Also holds for parameters within families
  - Note context specific independence between $\theta_{Y|x1}$ and $\theta_{Y|x0}$ when given both X's and Y's

Daphne Koller

Bayesian Estimation in BNs





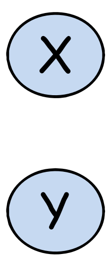
- Posteriors of θ can be computed independently
  - For multinomial $\theta_{X|\mathbf{u}}$ (assignment to X's parents $\mathbf{u}$) if prior is Dirichlet($\alpha_{x^1|\mathbf{u}},\ldots, \alpha_{x^k|\mathbf{u}}$)
  - posterior is Dirichlet($\alpha_{x^1|\mathbf{u}}+M[x^1,\mathbf{u}],\ldots,\alpha_{x^k|\mathbf{u}}+M[x^k,\mathbf{u}]$)

Daphne Koller

Assessing Priors for BNs

- We need hyperparameter $\alpha_{x|\mathbf{u}}$ for each node X, value x, and parent assignment **u**
  - Prior network with parameters Θ_0
  - Equivalent sample size parameter α
  - $\alpha_{x|\mathbf{u}} := \alpha \cdot P(x,\mathbf{u}|\Theta_0)$

$X = x, \bar{U} = \bar{u}$



Θ_0 uniform

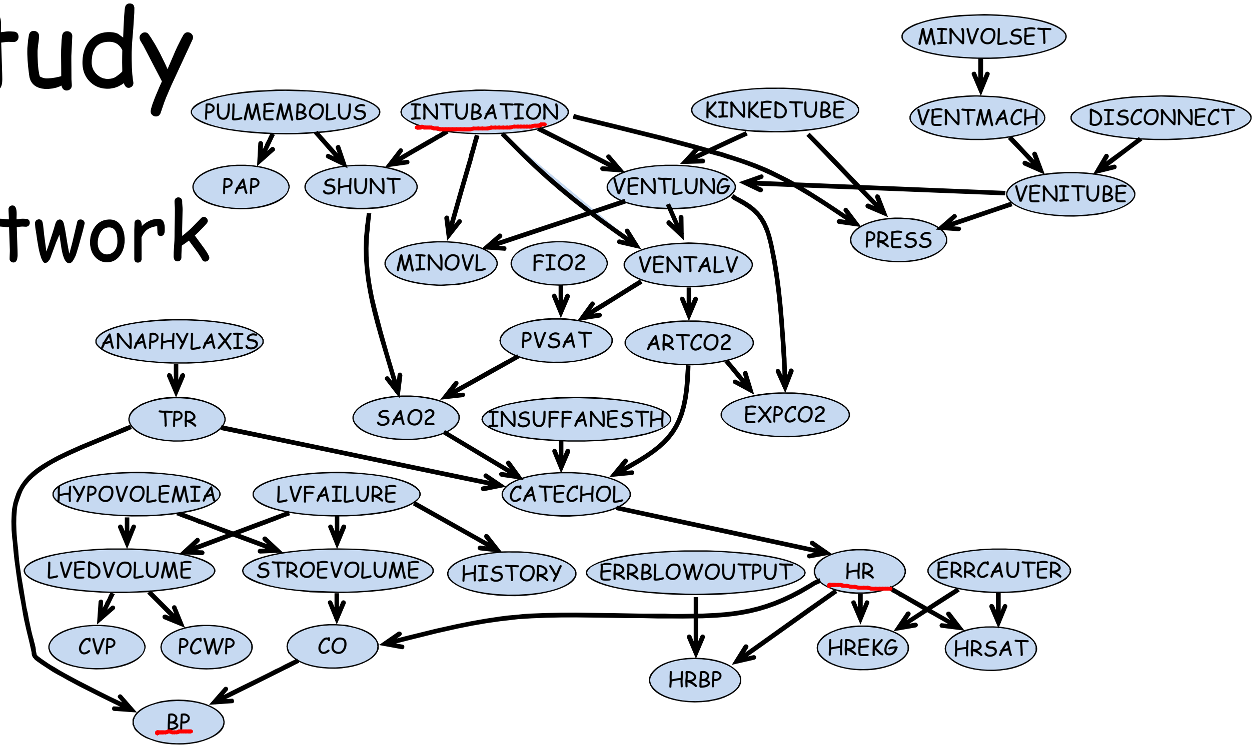
$X \quad \theta_X \sim \text{Dirichlet}(\frac{\alpha}{2}, \frac{\alpha}{2})$

$Y \quad \theta_{Y|x^0} \sim \text{Dirichlet}(\alpha/4, \alpha/4)$

$\theta_{Y|x^1} \sim \text{Dirichlet}(\alpha/4, \alpha/4)$

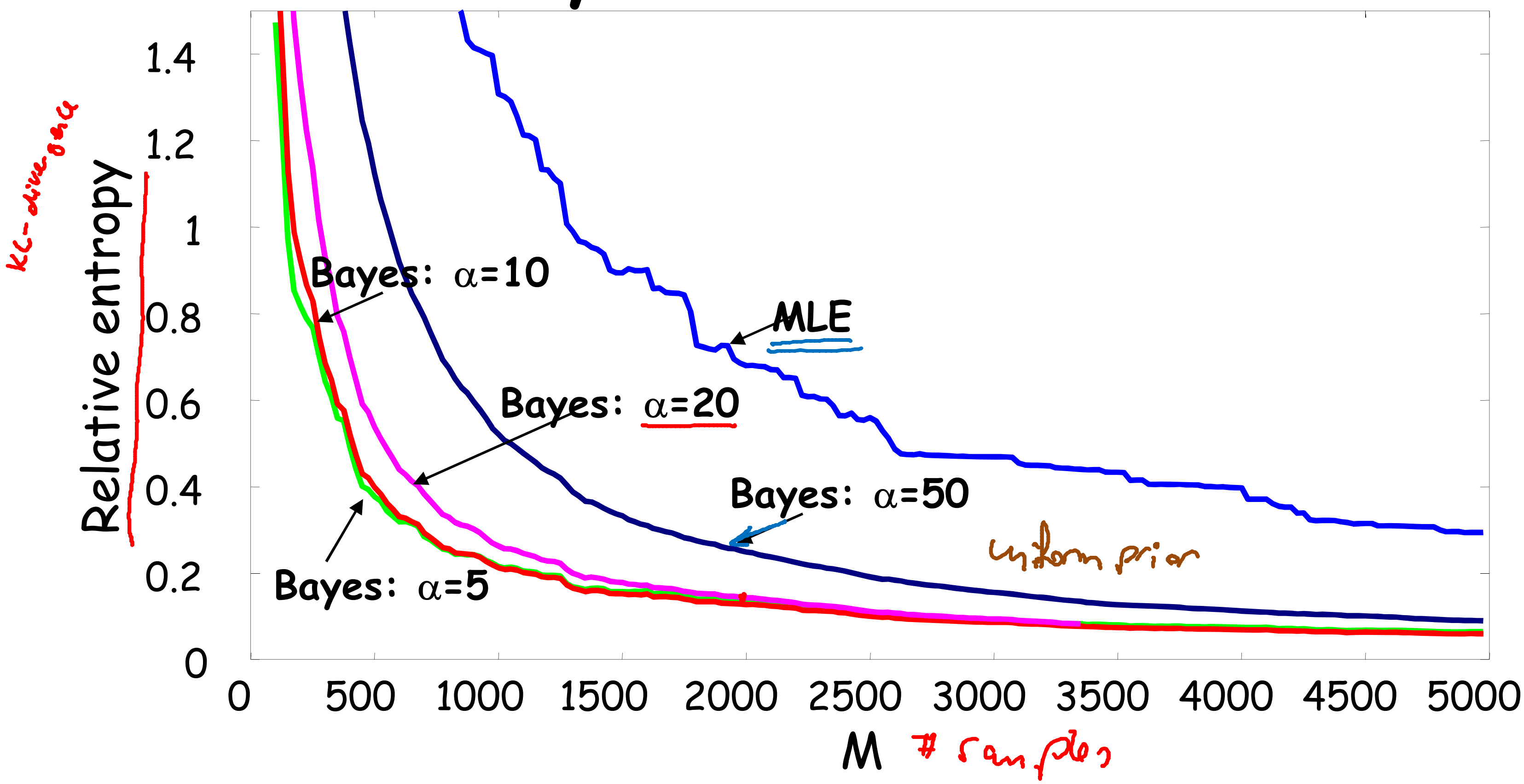
Case Study

- ICU-Alarm network
  - 37 variables
  - 504 params
- Experiment
  - Sample instances from network
  - Relearn parameters



Daphne Koller

Case Study: ICU Alarm Network



Daphne Koller

Summary

- In Bayesian networks, if parameters are independent a priori, then also independent in the posterior
- For multinomial BNs, estimation uses sufficient statistics M[x,**u**]

$$\hat{\theta}_{x|\boldsymbol{u}} = \frac{M[x, \boldsymbol{u}]}{M[\boldsymbol{u}]}$$

MLE

$$P(x \mid \boldsymbol{u}, D) = \frac{\alpha_{x,\boldsymbol{u}} + M[x, \boldsymbol{u}]}{\alpha_{\boldsymbol{u}} + M[\boldsymbol{u}]}$$

Bayesian (Dirichlet)

- Bayesian methods require choice of prior
  - can be elicited as prior network and equivalent sample size α

Daphne Koller