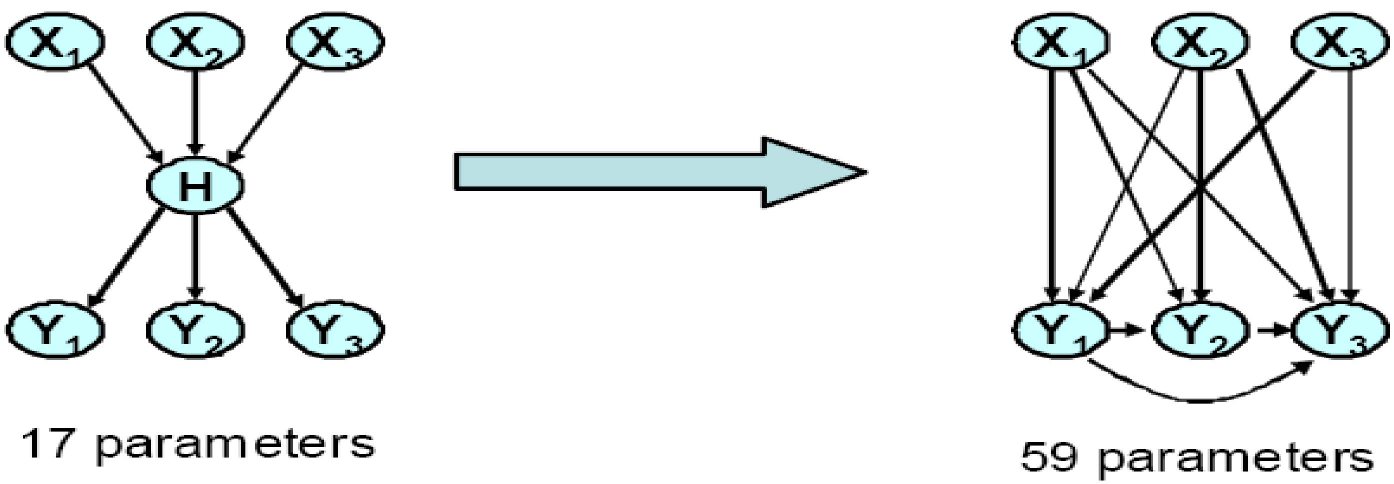
Stat 521A
Lecture 20

Outline

- Overview of learning (ch 16)
- MLE for DGMs (17.2)
- Bayesian parameter estimation for DGMs (17.4)
- Parameter tying (17.5)
- Hierarchical Bayes (17.5.4)
- PAC analysis (17.6)

Overview of learning

- Learn parameters or structure
- Observe all variables, or have missing values, or have known hidden variables, or have unknown hidden variables
- Hidden variables can simplify the model (fewer params)



Overview of learning

[t]

Learning Bayesian networks

	Complete data	Missing data	Hidden variables
Known structure	Closed form solution	• Iterated optimization to local maximum, • Inference on network multiple times	• Symmetrical solutions • Infinite # of solutions
Unknown structure, known variables	• Combinatorial optimization over structures • score has closed form	• Inference over multiple different network structures • no closed form for score	
Unknown vars	N/A	N/A	• Infinite number of possible solutions

Learning Markov networks

	Complete data	Missing data	Hidden variables
Known structure	• Convex optimization problem solved optimally via numerical optimization • Inference on network multiple times	• Non-convex problem • Iterated optimization to local maximum • Inference on network multiple times	• Symmetrical solutions • Infinite # of solutions
Unknown structure, known variables	• Combinatorial and numerical formulations • Can be solved via convex optimization • Inference over multiple different network structures		
Unknown vars	N/A	N/A	• Infinite number of possible solutions

Rest of ch 16

- Overfitting
- Cross validation
- Empirical risk minimization
- PAC bounds (see later)
- Generative vs discriminative
- Bias/variance tradeoff
- Prediction vs density estimation vs knowledge discovery

MLE for DGMs

- Assume DAG is known and variables are fully observed
- The likelihood factorizes into a product of local likelihoods, so we can optimize each CPD independently

$$
\begin{aligned}
p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} \prod_{i=1}^{D} p(x_{in}|\mathbf{x}_{\pi_i,n}, \boldsymbol{\theta}_i) \\
&= \prod_{i=1}^{D} [\prod_{n=1}^{N} p(x_{in}|\mathbf{x}_{\pi_i,n}, \boldsymbol{\theta}_i)] \\
&= \prod_{i=1}^{D} p(\mathcal{D}_i|\boldsymbol{\theta}_i)
\end{aligned}
$$

Tabular CPDs

$$\theta_{ijk} \stackrel{\text{def}}{=} p(X_i = k | \mathbf{X}_{\pi_i} = j)$$

$$\prod_{n=1}^{N} p(x_{in} | \mathbf{x}_{\pi_i,n}, \boldsymbol{\theta}_i) = \prod_{n=1}^{N} \prod_{j=1}^{r_i} \prod_{k=1}^{q_i} \theta_{ijk}^{I(x_{i,n}=k, \mathbf{x}_{\pi_i,n}=j)}$$

$$= \prod_{j} \prod_{k} \theta_{ijk}^{N_{ijk}}$$

$$N_{ijk} \stackrel{\text{def}}{=} \sum_{n=1}^{N} I(x_{i,n} = k, \mathbf{x}_{\pi_i,n} = j)$$

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{j'=1}^{r_i} N_{ij'k}}$$

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

MLE for linear Gaussian CPDs

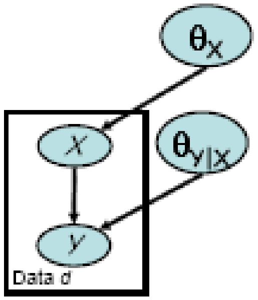
- Use usual linear regression equations

$$p(x_i|\mathbf{x}_{\pi_i}, \boldsymbol{\theta}_i) \quad = \quad \mathcal{N}(x_i|\mathbf{w}_i^T \mathbf{x}_{\pi_i}, \sigma_i^2)$$

Bayesian parameter estimation

- Global parameter independence

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{D} p(\boldsymbol{\theta}_i)$$



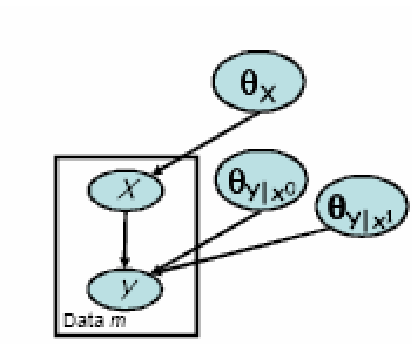
- Implies factorized posterior

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \prod_i p(\boldsymbol{\theta}_i) p(\mathcal{D}_i|\boldsymbol{\theta}_i)$$

- For multinomials, let us assume local param indep

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{r_i} p(\boldsymbol{\theta}_{ij})$$

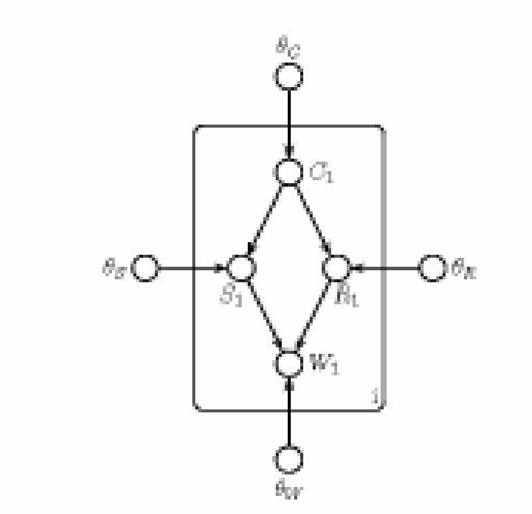
- Geiger & Heckerman showed this implies theta_{ij} must have a Dirichlet prior



Tabular CPDs

- We have

$$p(\boldsymbol{\theta}|\mathcal{D}) \quad \propto \quad \prod_{i=1}^{D}\prod_{j=1}^{r_i} \text{Dir}(\boldsymbol{\theta}_{ij}|\boldsymbol{\alpha}_{ij} + \mathbf{N}_{ij})$$

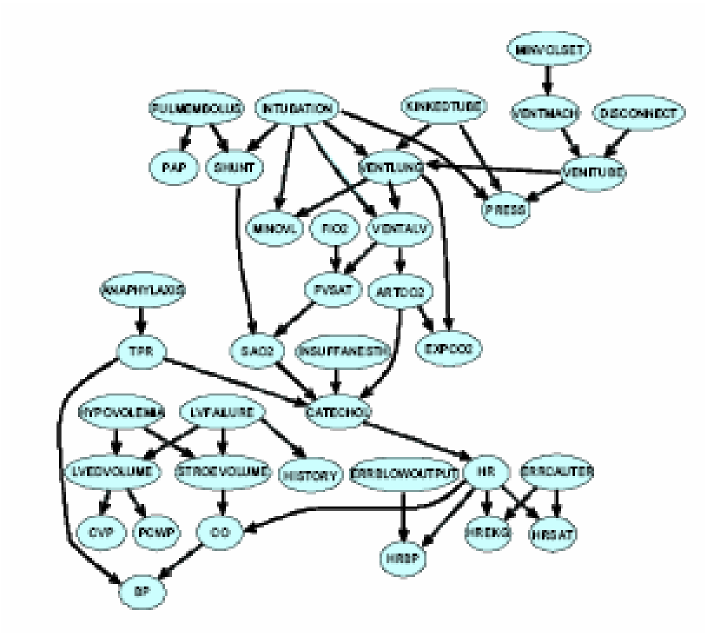


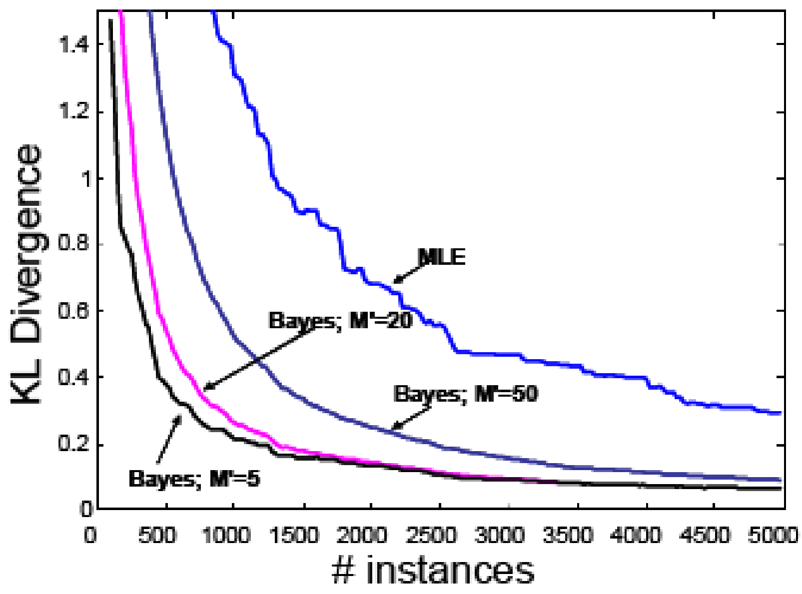
i	C	S	R	W
1	0	0	0	0
2	0	0	1	1
3	1	1	1	1

$p(\theta_C)$		$p(\theta_R \mid C=0)$		$p(\theta_R \mid C=1)$	
1	1	1	1	1	1
2	1	2	1	1	1
3	1	2	2	1	1
3	2	2	2	1	2

Example

- ICU alarm network, 37 nodes, 504 params
- Compute theta-hat using MLE or posterior mean. Then compute KL(p(X|theta*), p(X|theta-hat)) as a function of sample size.

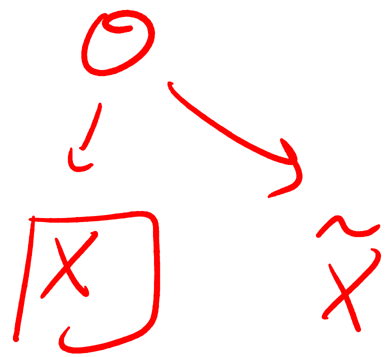




Posterior predictive density

- We can predict future variables by integrating out the params

$$p(\tilde{X}|\mathcal{D}) = \int p(\tilde{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$



- In the case of Dirichlet-multinomial model, this is equivalent to plugging in the posterior mean

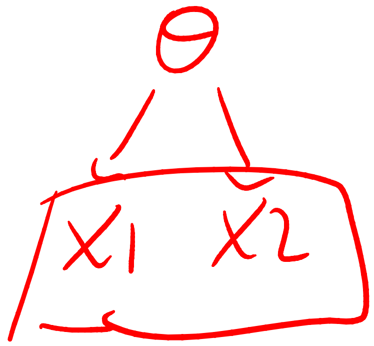
$$\begin{aligned} p(\tilde{X} = k|\mathcal{D}) &= \int \theta_k p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \\ &= \int \theta_k p(\boldsymbol{\theta}_k|\mathcal{D})d\boldsymbol{\theta} \\ &= \overline{\theta}_k = \frac{\alpha_k + N_k}{\sum_{k'} \alpha_{k'} + N_{k'}} \end{aligned}$$

MAP estimation

- Since in general computing the posterior is difficult, a compromise is to compute a MAP estimate
- However, the result is not invariant to parameterization – change of variables formula changes the prior density (Box 17.D)
- Reparameterizing the likelihood does not change the MLE, since the lik is not a density function
- Reparameterizing the posterior does not change anything, since we integrate over params

Parameter tying

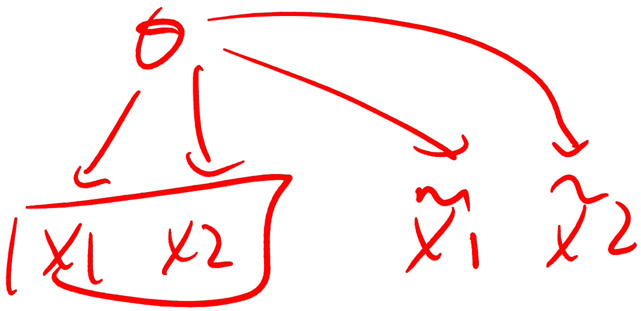
- We just pool the sufficient statistics from the nodes that share the same params



$$p(\boldsymbol{\theta}|\mathcal{D}) = \text{Dir}(\alpha_k + \sum_n I(X_{1n} = j) + \sum_n I(X_{2n} = j))$$

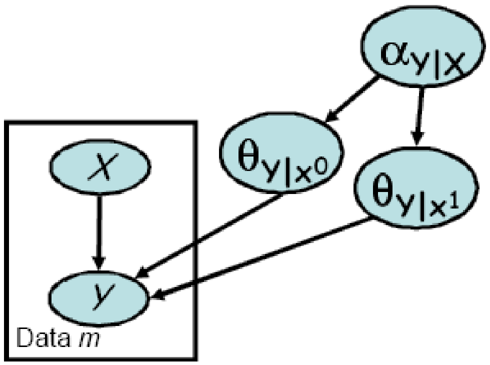
Prediction with tied params

- A subtlety arises when computing the posterior predictive density with tied params
- When we observe Xtilde1, we learn something about theta that helps us predict Xtilde2. So we cannot just multiply the postpred for each node separately, but need to use the formula for a batch of data

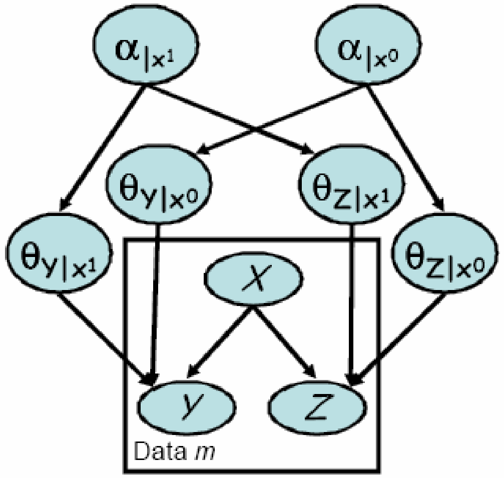


Hierarchical priors

- Encourage params to be similar across conditioning contexts (rows) within 1 CPD

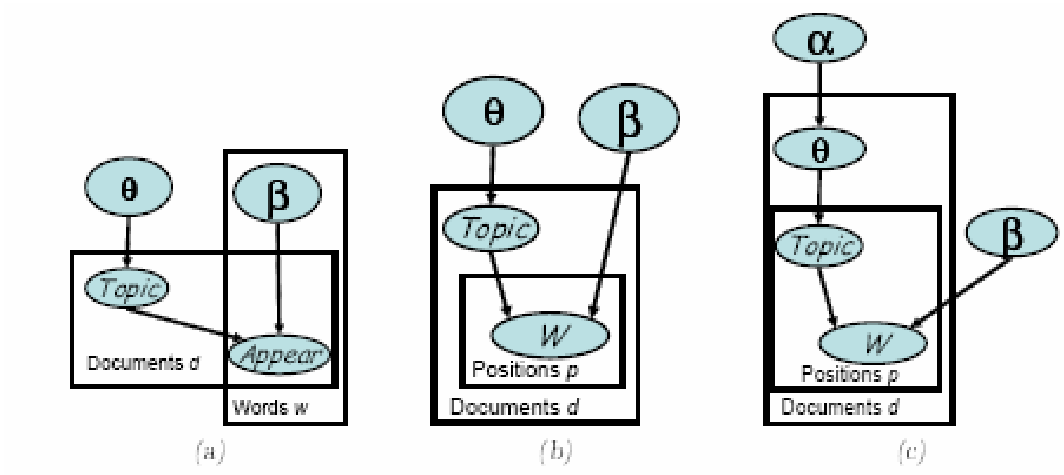


- Encourage params to be similar across response for each conditionign context



Text classification

- Let T(d) = t be topic of document d, ~Mun(θ)
- Product of Bernoullis, A(w,d) in {0,1} ~ Ber($\beta_{w,t}$), w =1:K
- Product of multinoullis, W(p,d) in {1,…,K} ~ Mun(β_t), p=1:len(d)
- Latent Dirichlet Allocation: θ(d) = distribution over topics, ~Dir(α), T(p,d) = t, W(p,d) ~ Mun(β_t)



PAC analysis

- Probably approximately correct
- Let P*M be distribution over datasets of size M drawn from P*
- $P_{ML(D)}$ be distribution over X given by model M learned using algo L on data D
- We want to prove that

Let $\epsilon > 0$ be our approximation parameter and $\delta > 0$ our confidence parameter. Then, for M "large enough", we have that

$$P^*_M(\{\mathcal{D} : D(P^* \| P_{\mathcal{M}_{L(\mathcal{D})}}) \leq \epsilon\}) \geq 1 - \delta.$$

- Frequentist analysis of estimator; bounds on deviation from 'truth'

Excess risk

- Minimizing KL(P*,P) may be hard if P* is not in the model class of P
- Define best achievable param in class as

$$\theta^{\text{opt}} = \arg \min_{\theta \in \Theta[\mathcal{G}]} D(P^* \| P_\theta).$$

- Define excess risk as

$$D(P^* \| P_\theta) - D(P^* \| P_{\theta^{\text{opt}}}) :$$

DGM param learning: PAC bounds

Theorem 17.6.8: *Let $\mathcal{G}$ be a network structure, and P^* a distribution consistent with some network $\mathcal{G}^*$ such that $P^*(x_i \mid \mathrm{pa}_i^{\mathcal{G}^*}) \geq \lambda$ for all i, x_i, and $\mathrm{pa}_i^{\mathcal{G}^*}$. If P is the distribution learned by maximum likelihood estimate for $\mathcal{G}$, then*

$$P\left(\boldsymbol{D}(P^* \| P) - \boldsymbol{D}(P^* \| P_{\boldsymbol{\theta}^{\mathrm{opt}}}) > n\epsilon\right) \leq nK^{d+1} e^{-2M\lambda^{2(d+1)}\epsilon^2 \frac{1}{(1+\epsilon)^2}}$$

where K is the maximal variable cardinality and d is the maximum number of parents in $\mathcal{G}$.

Corollary 17.6.9: *Under the conditions of Theorem 17.6.8, if*

$$M \geq \frac{1}{2} \frac{1}{\lambda^{2(d+1)}} \frac{(1+\epsilon)^2}{\epsilon^2} \log \frac{nK^{d+1}}{\delta},$$

then

$$P\left(\boldsymbol{D}(P^* \| P) - \boldsymbol{D}(P^* \| P_{\boldsymbol{\theta}^{\mathrm{opt}}}) < n\epsilon\right) > 1 - \delta.$$