Readings:
K&F: 14.1, 14.2, 14.3, 14.4, 15.1, 15.2, 15.3.1, 15.4.1

Parameter Learning 2

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

September 27th, 2006

Your first learning algorithm

$$\ln a^b = b \ln a$$
$$\ln a \cdot b = \ln a + \ln b$$
$$\frac{\partial}{\partial\theta} \ln\theta = \frac{1}{\theta}$$
$$\frac{\partial}{\partial\theta} \ln(1-\theta) = \frac{-1}{1-\theta}$$

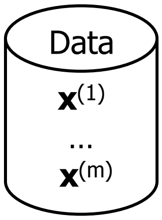
$$\begin{aligned}\widehat{\theta} &= \arg\max_\theta \quad \ln P(\mathcal{D} \mid \theta) \\ &= \arg\max_\theta \quad \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}\end{aligned}$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) = 0}$

$$\frac{\partial}{\partial\theta} \ln\left[\theta^{\alpha_H}(1-\theta)^{\alpha_T}\right]$$
$$= \frac{\partial}{\partial\theta}\left[\alpha_H \ln\theta + \alpha_T \ln(1-\theta)\right] = \frac{\partial}{\partial\theta} \alpha_H \ln\theta + \frac{\partial}{\partial\theta} \alpha_T \ln(1-\theta)$$
$$= \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1-\theta} = 0 \quad \Rightarrow \quad \theta = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

10-708 – ©Carlos Guestrin 2006

Learning the CPTs

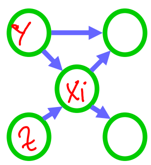


For each discrete variable X_i

$$P(X_i \mid Pa_{X_i}) = P(X_i \mid Y, Z)$$

$$\overset{MLE}{\approx}$$

$$P(X_i = x_i \mid Y=y, Z=z) \overset{MLE}{\approx} \frac{Count(X_i=x_i, Y=y, Z=z)}{Count(Y=y, Z=z)}$$



MLE: $P(X_i = x_i \mid X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$

Maximum likelihood estimation (MLE) of BN parameters – General case

- Data: $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}$
- Restriction: $\mathbf{x}^{(j)}[\mathbf{Pa}_{X_i}] \rightarrow$ assignment to $\mathbf{Pa}_{X_i}$ in $\mathbf{x}^{(j)}$
- Given structure, log likelihood of data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \log \prod_j \prod_i P(X_i = x_i^{(j)} \mid Pa_{X_i} = x^{(j)}[Pa_{X_i}])$$

Taking derivatives of MLE of BN parameters – General case

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P\left(X_i = x_i^{(j)} \mid \mathbf{Pa}_{X_i} = \mathbf{x}^{(j)}\left[\mathbf{Pa}_{X_i}\right]\right)$$

General MLE for a CPT

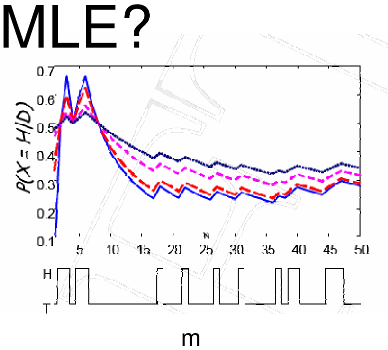
- Take a CPT: P(X|**U**)
- Log likelihood term for this CPT
- Parameter $\theta_{X=x|\mathbf{U}=\mathbf{u}}$:

MLE: $P(X = x \mid \mathbf{U} = \mathbf{u}) = \theta_{X=x|\mathbf{U}=\mathbf{u}} = \dfrac{\text{Count}(X = x, \mathbf{U} = \mathbf{u})}{\text{Count}(\mathbf{U} = \mathbf{u})}$

Announcements

- Late homeworks:
  - 3 late days for the semester
    - one late day corresponds to 24 hours! (i.e., 3 late days due Saturday by noon)
    - Give late homeworks to Monica Hopes, Wean Hall 4619
      - If she is not in her office, time stamp (date and time) your homework, sign it, and put it under her door
  - After late days are used up:
    - Half credit within 48 hours
    - Zero credit after 48 hours
  - All homeworks **must be handed in**, even for zero credit
- Homework 2 out later today
- Recitation tomorrow:
  - review perfect maps, parameter learning

Can we really trust MLE?



- What is better?
  - 3 heads, 2 tails
  - 30 heads, 20 tails
  - $3\text{x}10^{23}$ heads, $2\text{x}10^{23}$ tails
- Many possible answers, we need distributions over possible parameters

Bayesian Learning

- Use Bayes rule:

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})}$$

- Or equivalently:

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta)$$

Bayesian Learning for Thumbtack

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta)$$

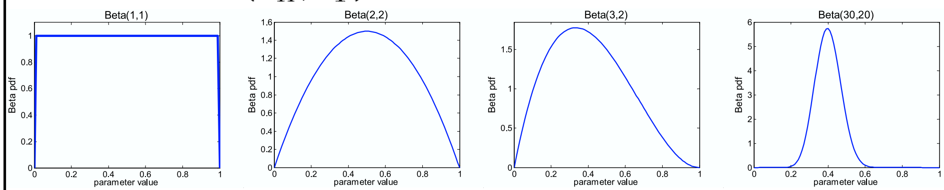
- Likelihood function is simply Binomial:

$$P(\mathcal{D} \mid \theta) = \theta^{m_H}(1-\theta)^{m_T}$$

- What about prior?
  - Represent expert knowledge
  - Simple posterior form
- Conjugate priors:
  - Closed-form representation of posterior (more details soon)
  - **For Binomial, conjugate prior is Beta distribution**

Beta prior distribution – P(θ)

$$P(\theta) = \frac{\theta^{\alpha_H - 1}(1-\theta)^{\alpha_T - 1}}{B(\alpha_H, \alpha_T)} \sim Beta(\alpha_H, \alpha_T)$$

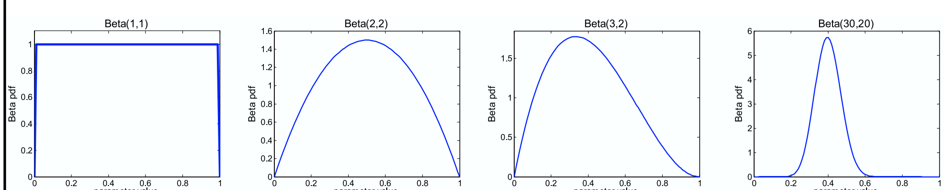


- Likelihood function: $P(\mathcal{D} \mid \theta) = \theta^{m_H}(1-\theta)^{m_T}$
- Posterior: $P(\theta \mid \mathcal{D}) \ \propto \ P(\mathcal{D} \mid \theta)P(\theta)$

Posterior distribution

- Prior: $Beta(\alpha_H, \alpha_T)$
- Data: m_H heads and m_T tails
- Posterior distribution:

$$P(\theta \mid \mathcal{D}) \sim Beta(m_H + \alpha_H, m_T + \alpha_T)$$



Conjugate prior

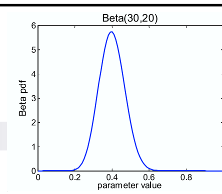
- Prior: $Beta(\alpha_H, \alpha_T)$
- Data: m_H heads and m_T tails (binomial likelihood)
- Posterior distribution:

$$P(\theta \mid \mathcal{D}) \sim Beta(m_H + \alpha_H, m_T + \alpha_T)$$

- Given likelihood function P(*D*|θ)

- (Parametric) prior of the form P(θ|α) is **conjugate** to likelihood function if posterior is of the same parametric family, and can be written as:
  - P(θ|α'), for some new set of parameters α'

Using Bayesian posterior



- Posterior distribution:

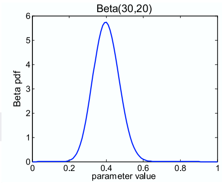
$$P(\theta \mid \mathcal{D}) \sim Beta(m_H + \alpha_H, m_T + \alpha_T)$$

- Bayesian inference:
  - No longer single parameter:

$$E[f(\theta)] = \int_0^1 f(\theta) P(\theta \mid \mathcal{D}) d\theta$$

  - Integral is often hard to compute

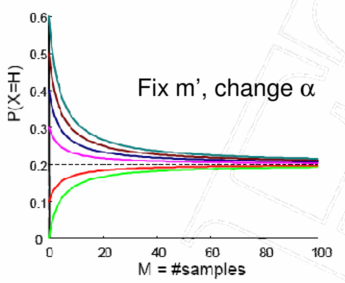
Bayesian prediction of a new coin flip

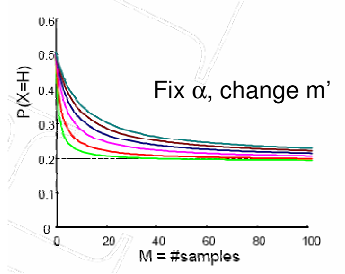


- Prior:
- Observed m_H heads, m_T tails, what is probability of m+1 flip is heads?

Asymptotic behavior and equivalent sample size

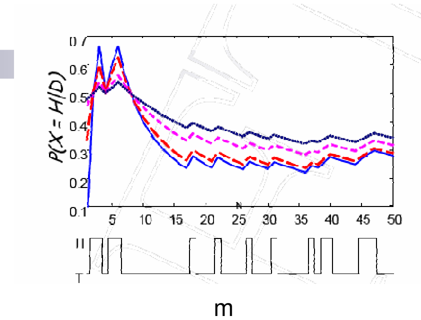
- Beta prior equivalent to extra thumbtack flips:
  - $$E[\theta] = \frac{m_H + \alpha_H}{m_H + \alpha_H + m_T + \alpha_T}$$
- As $m \to \infty$, prior is "forgotten"
- **But, for small sample size, prior is important!**
- **Equivalent sample size**:
  - Prior parameterized by α_H, α_T, or
  - m' (equivalent sample size) and α
  - $$E[\theta] = \frac{m_H + \alpha m'}{m_H + m_T + m'}$$





Bayesian learning corresponds to smoothing

$$E[\theta] = \frac{m_H + \alpha m'}{m_H + m_T + m'}$$



- m=0 ⇒ prior parameter
- m→∞ ⇒ MLE

Bayesian learning for multinomial

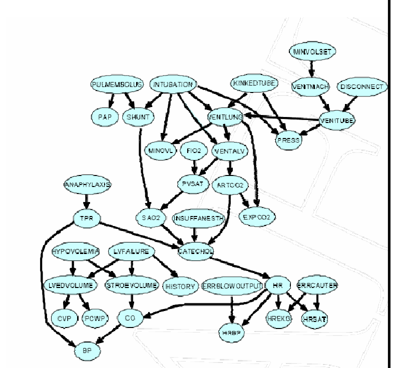
- What if you have a k sided coin???
- Likelihood function if **multinomial**:
  -
  -
- **Conjugate** prior for multinomial is **Dirichlet**:
  - $\theta \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k) \sim \prod_i \theta_i^{\alpha_i - 1}$
- **Observe** *m* data points, m_i from assignment i, **posterior**:
- **Prediction**:

Bayesian learning for two-node BN

- Parameters θ_X, $\theta_{Y|X}$
- Priors:
  - $P(\theta_X)$:
  - $P(\theta_{Y|X})$:

Very important assumption on prior: Global parameter independence

- **Global parameter independence:**
  - Prior over parameters is product of prior over CPTs

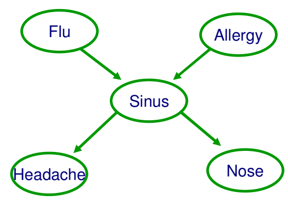


Global parameter independence, d-separation and local prediction

- Independencies in **meta BN**:

- **Proposition**: For fully observable data *D*, if prior satisfies global parameter independence, then

$$P(\theta \mid \mathcal{D}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}} \mid \mathcal{D})$$



Within a CPT

- Meta BN including CPT parameters:

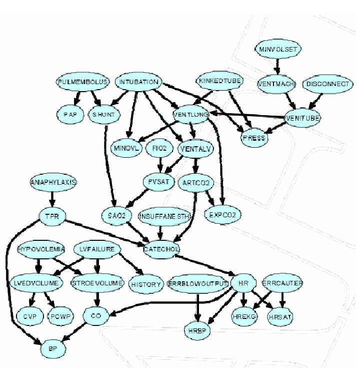
- Are $\theta_{Y/X=t}$ and $\theta_{Y|X=f}$ d-separated given *D*?
- Are $\theta_{Y|X=t}$ and $\theta_{Y|X=f}$ independent given *D*?
  - Context-specific independence!!!
- Posterior decomposes:

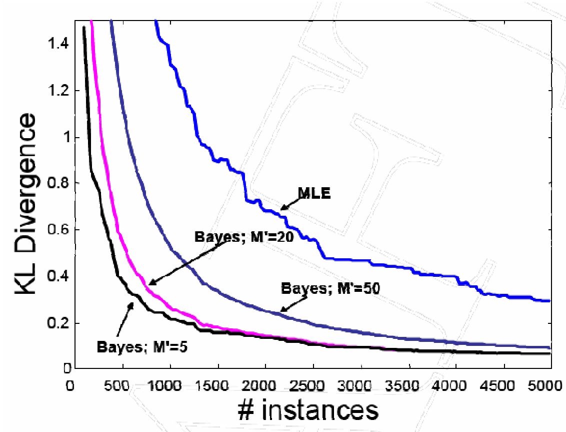
Priors for BN CPTs

(more when we talk about structure learning)

- Consider each CPT: P(X|**U**=**u**)
- Conjugate prior:
  - Dirichlet($\alpha_{X=1|\mathbf{U}=\mathbf{u}}, \ldots, \alpha_{X=k|\mathbf{U}=\mathbf{u}}$)
- More intuitive:
  - "prior data set" *D*' with m' equivalent sample size
  - "prior counts":
  - prediction:

An example





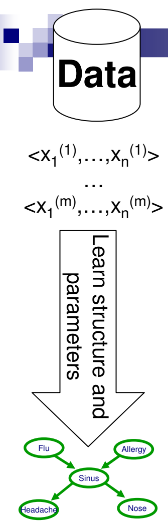
What you need to know about parameter learning

- MLE:
  - score decomposes according to CPTs
  - optimize each CPT separately
- Bayesian parameter learning:
  - motivation for Bayesian approach
  - Bayesian prediction
  - conjugate priors, equivalent sample size
  - Bayesian learning $\Rightarrow$ smoothing
- Bayesian learning for BN parameters
  - Global parameter independence
  - Decomposition of prediction according to CPTs
  - Decomposition within a CPT

Where are we with learning BNs?

- Given structure, estimate parameters
  - Maximum likelihood estimation
  - Bayesian learning
- What about learning structure?

Learning the structure of a BN



- **Constraint-based approach**
  - BN encodes conditional independencies
  - Test conditional independencies in data
  - Find an I-map
- **Score-based approach**
  - Finding a structure and parameters is a density estimation task
  - Evaluate model as we evaluated parameters
    - Maximum likelihood
    - Bayesian
    - etc.

Remember: Obtaining a P-map?

- Given the independence assertions that are true for *P*
  - Obtain skeleton
  - Obtain immoralities
- From skeleton and immoralities, obtain every (and any) BN structure from the equivalence class

- **Constraint-based approach**:
  - Use Learn PDAG algorithm
  - Key question: **Independence test**

Independence tests

- Statistically difficult task!
- Intuitive approach: **Mutual information**

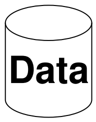
$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

- Mutual information and independence:
  - X_i and X_j independent if and only if $I(X_i,X_j)=0$
- Conditional mutual information:

Independence tests and the constraint based approach

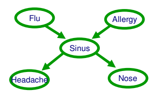
- Using the data *D*
  - Empirical distribution: $\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$
  - Mutual information: $\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$
  - Similarly for conditional MI
- Use learning PDAG algorithm:
  - When algorithm asks: $(X \perp Y | \mathbf{U})$?
- Many other types of independence tests
  - See reading…

Score-based approach



$<x_1^{(1)},\ldots,x_n^{(1)}>$
…
$<x_1^{(m)},\ldots,x_n^{(m)}>$

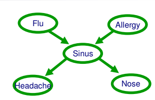
Possible structures



Learn parameters

Score structure

Information-theoretic interpretation of maximum likelihood

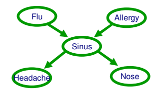


- Given structure, log likelihood of data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P\left(X_i = x_i^{(j)} \mid \mathbf{Pa}_{X_i} = \mathbf{x}^{(j)}\left[\mathbf{Pa}_{X_i}\right]\right)$$

Information-theoretic interpretation of maximum likelihood 2

- Given structure, log likelihood of data:



$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i,\mathcal{G}}} \hat{P}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) \log \hat{P}(x_i \mid \mathbf{Pa}_{x_i,\mathcal{G}})$$

Decomposable score

- Log data likelihood

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - Score(G : D) = $\sum_i$ FamScore($X_i|\mathbf{Pa}_{X_i}$: D)

How many trees are there?

Nonetheless – Efficient optimal algorithm finds best tree

Scoring a tree 1: I-equivalent trees

$$\log \widehat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \widehat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \widehat{H}(X_i)$$

Scoring a tree 2: similar trees

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

Chow-Liu tree learning algorithm 1

- For each pair of variables X_i,X_j
  - Compute empirical distribution:

$$\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$

  - Compute mutual information:

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i)\hat{P}(x_j)}$$

- Define a graph
  - Nodes $X_1,\ldots,X_n$
  - Edge (i,j) gets weight $\hat{I}(X_i, X_j)$

Chow-Liu tree learning algorithm 2

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

- Optimal tree BN
  - Compute maximum weight spanning tree
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Can we extend Chow-Liu 1

- Tree augmented naïve Bayes (TAN) [Friedman et al. '97]
  - Naïve Bayes model overcounts, because correlation between features not considered
  - Same as Chow-Liu, but score edges with:

$$\hat{I}(X_i, X_j \mid C) = \sum_{c, x_i, x_j} \hat{P}(c, x_i, x_j) \log \frac{\hat{P}(x_i, x_j \mid c)}{\hat{P}(x_i \mid c)\hat{P}(x_j \mid c)}$$

Can we extend Chow-Liu 2

- (Approximately learning) models with tree-width up to *k*
  - [Narasimhan & Bilmes '04]
  - But, $O(n^{k+1})$…
    - and more subtleties

What you need to know about learning BN structures so far

- Decomposable scores
  - Maximum likelihood
  - Information theoretic interpretation
- Best tree (Chow-Liu)
- Best TAN
- Nearly best k-treewidth (in $O(N^{k+1})$)