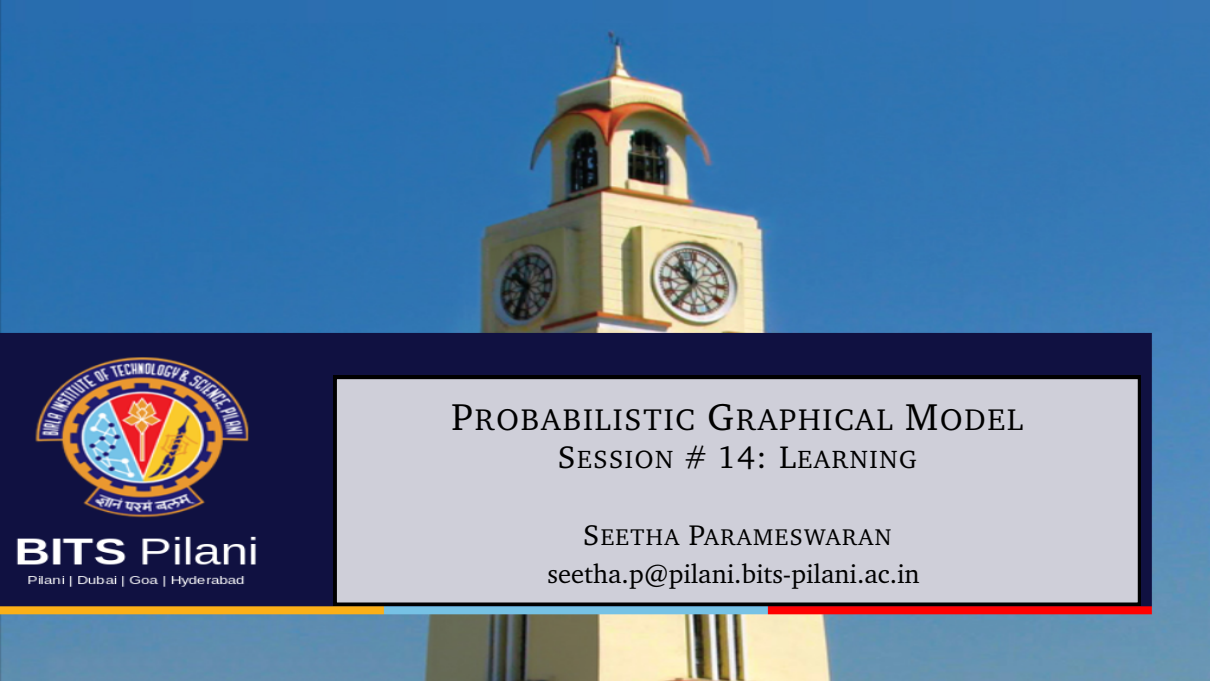


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Probabilistic Graphical Model

Session # 14: Learning

Seetha Parameswaran
seetha.p@pilani.bits-pilani.ac.in

Table of Contents

Structure Learning in Bayesian Networks

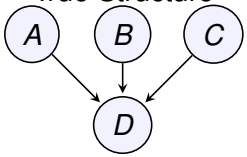
- We do not know the structure of the Bayesian network in advance.
- The data are generated IID from an underlying distribution $P^*(\mathcal{X})$.
- $P^*(\mathcal{X})$ is induced by some Bayesian network G^* over $\mathcal{X}$.

Why Structure Learning?

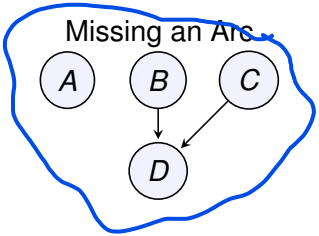
- To learn model for new queries when the domain expert is not perfect.
- For structure discovery, when inferring network structure is goal in itself.

Structure – Which one?

True Structure

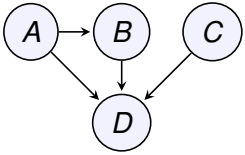


Missing an Arc



- Incorrect independencies
- Correct P^* cannot be learned
- Could generalize better

Adding an Arc

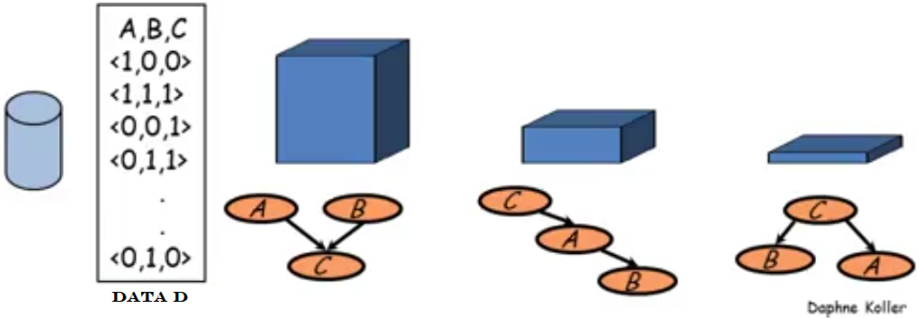


- Spurious dependencies
- Cannot correctly learn P^*
- More number of parameters are learned
- Worse generalization

Table of Contents

Score-based Learning

1. Define **scoring function** that evaluates how well a structure matches the data.



2. Search for a structure that maximizes the score. This converts the learning problem into an optimization problem.

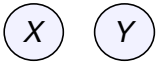
Likelihood Structure Score

- Simplest score
- Find the graph and parameters that maximize the likelihood of the data.

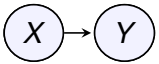
$$score_L(\mathcal{G} : \mathcal{D}) = \ell((\hat{\theta}, \mathcal{G}) : \mathcal{D})$$

- $\hat{\theta} =$ MLE of parameters given $\mathcal{G}$ and $\mathcal{D}$

Example

$\mathcal{G}_0$ 

$$score_L(\mathcal{G}_0 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]})$$

$\mathcal{G}_1$ 

$$score_L(\mathcal{G}_1 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]})$$

Example

$$score_L(\mathcal{G}_1 : \mathcal{D}) - score_L(\mathcal{G}_0 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{y[m]|x[m]}) - \sum_m (\log \hat{\theta}_{y[m]})$$

$$= \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y$$

sufficient statistics

$$= M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - M \sum_y \hat{P}(y) \log \hat{P}(y)$$

empirical distribution $\hat{P}(x,y)$

$$= M \left(\sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y) \right)$$

$$\sum_x \hat{P}(x,y) = \hat{P}(y)$$

$$= M \sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)}$$

$$\hat{P}(y \mid x) = \frac{\hat{P}(x,y)}{\hat{P}(x)}$$

$$= M \cdot \mathrm{I}_{\hat{P}}(X;Y)$$

samples times Mutual Information

Likelihood Structure Score

- In general, the Likelihood Structure Score decomposes as number of samples M times the difference between the mutual information and the sum of entropies of variables X.

$$score_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \mathrm{I}_{\hat{P}}(X_i; Pa_{X_i}^{\mathcal{G}}) - M \sum_{i} \mathrm{H}_{\hat{P}}(X_i)$$

$$\mathrm{I}_{\hat{P}}(X; Y) = \sum_{x,y} \hat{P}(x, y) \log \frac{\hat{P}(x, y)}{\hat{P}(x)\hat{P}(y)}$$

$$\mathrm{H}_{\hat{P}}(X) = -\sum_{x} P(x) \log P(x)$$

- The score is higher if X_i is correlated with its parents. Helps in identifying the Parent-Child relationship and place the nodes accordingly.

Limitations of Likelihood Score

$$score_L(\mathcal{G}_1 : \mathcal{D}) - score_L(\mathcal{G}_0 : \mathcal{D}) = M \cdot \mathrm{I}_{\hat{P}}(X; Y)$$

- Positive difference suggests that $\mathcal{G}_1$ has to be chosen.
- Negative difference implies that $\mathcal{G}_0$ has to be chosen.
- Mutual Information is always positive. $\mathrm{I}_{\hat{P}}(X; Y) \geq 0$
- Mutual Information $\mathrm{I}_{\hat{P}}(X; Y) = 0$ iff X and Y are independent.
- In empirical distribution $\hat{P}$, due to statistical fluctuations while taking samples, $\mathrm{I}_{\hat{P}}(X; Y) > 0$ almost always.
- Score is maximized for fully connected network.
- Over-fits the data

why is this always true?

Why is mutual information always non-negative?

$$I(X,Y) = -\sum_x \sum_y p(x,y) \log \frac{p(x)p(y)}{p(x,y)}$$

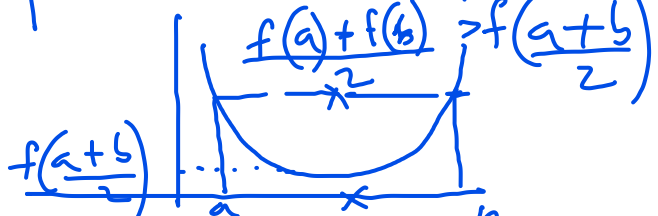
$\log$ is a concave function since

$$\frac{d^2}{dx^2} \log x = -\frac{1}{x^2} < 0$$

: $-\log$ is a convex function

We can apply Jensen's inequality on $-\log$, which is a convex function

$$E(f(X)) \geq f(E(X))$$ where X is a random variable, and f is a convex function, E is expectation over f of X



$$I(X,Y) \geq -\log \sum_x \sum_y p(x,y) \frac{p(x)p(y)}{p(x,y)}$$

$$\geq -\log \sum_x \sum_y p(x)p(y)$$

$$\geq -\log \left(\left(\sum_x p(x) \right) \left(\sum_y p(y) \right) \right)$$

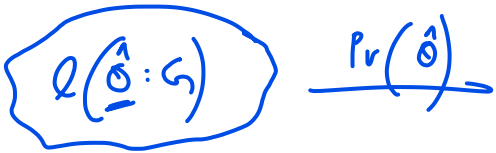
$$\geq -\log 1 \geq 0$$

Avoid Over-fitting

- Restrict the hypothesis space
  - Restrict # parents – common strategy
  - Restrict # parameters
- Score that penalize complexity
  - Explicitly penalize model complexity
  - Use Bayesian score that averages over all possible parameter values.

Table of Contents

Bayesian Score



- Bayesian score is derived from Bayesian paradigm – Any uncertainty can be represented by using a probability distribution for it.
- Uncertainty over graph can be represented using a prior over graph.
- Uncertainty over parameters can be represented using a prior over parameters.

Bayesian Score

- Optimization problem – find a graph that maximizes a probability of $\mathcal{G}$ over $\mathcal{D}$

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

$P(\mathcal{D} \mid \mathcal{G})$ – Marginal likelihood
$P(\mathcal{G})$ – Prior over structures
$P(\mathcal{D})$ – Marginal probability of Data – normalizing constant (ignored)

Bayesian Score

- Taking log on both sides, Bayesian Score

$$score_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} \mid \mathcal{G}) + \log P(\mathcal{G})$$

$score_B(\mathcal{G} : \mathcal{D})$ – Bayesian score
$\log P(\mathcal{D} \mid \mathcal{G})$ – Marginal likelihood of Data given graph $\mathcal{G}$
$\log P(\mathcal{G})$ – Prior over structures – less significant

Marginal Likelihood of Data given Graph

- As $M \to \infty$ a network with Dirichlet prior satisfies

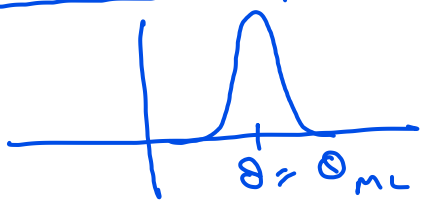
$$\log P(\mathcal{D} \mid \mathcal{G}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} Dim[\mathcal{G}] + \mathcal{O}(1)$$

- Marginal likelihood is the Bayesian Information Criterion (BIC) score.
- $\ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D})$ – Log likelihood score with Maximum likelihood parameters $\hat{\theta}_{\mathcal{G}}$
- M – # training data instances
- $Dim[\mathcal{G}]$ – # independent parameters = # entries in the distribution - 1

$$Pr(\theta|G) = \int Pr(\theta|\theta, G)\, Pr(\theta|G)\, d\theta \qquad \int Pr(\theta, \theta, G)\, d\theta$$

$Pr(\theta, G)$

$Pr(\theta|\theta, G)\, Pr(\theta|G)$ can be approximated as a function that peaks at $\theta = \theta_{ML} =$ max likelihood parameters



We can use second-order approximation (Taylor's series) and with

$$\log\left(Pr(D|\theta,G)Pr(\theta|G)\right) = \log\left(Pr(D|\theta_{MC},G)Pr\left(\frac{\theta_{MC}}{G}\right)\right) - \frac{1}{2}(\theta-\theta_{MC})^T H(\theta-\theta_{MC})$$

$$g(\theta) = g(\theta_{MC})e^{-\frac{1}{2}(\theta-\theta_{MC})^T H(\theta-\theta_{MC})}$$

where $H = \left[\frac{\partial^2 \ln g(\theta)}{\partial \theta \partial \theta^T} \right]$ a $d \times d$ matrix

$P(\theta/G) = \int g(\theta) d\theta$ is a Gaussian

integral can be approximated as

$$\frac{(2\pi)^{d/2}}{|H|^{\frac{1}{2}}} P_r(\theta/\theta_{MLE}, G) P_r(\theta_{MC}/G)$$

Taking $\ln P(\theta/G)$ and approximate $|H| = M^d$
+ get the result.

More specifically $\int_{-\infty}^{\infty} e^{-\frac{1}{2}x^T A x} d^n x \longrightarrow \int e^{-\frac{1}{2}(x-\mu)^T C (x-\mu)}$

$$= \frac{(2\pi)^{n/2}}{|A|^{\frac{1}{2}}}$$

Here $A = H$ and $n = d$, and we have a constant $Pr(D/\Theta_{MC}, G) Pr(\Theta_{MC}/G)$ in the integrand leading to the integral being equal to $\frac{(2\pi)^{d/2}}{|H|^{1/2}} Pr(D/\Theta_{MC}, G) Pr(\Theta_{MC}/G)$

$$Pr(D/G) = \frac{(2\pi)^{d/2}}{|H|^{1/2}} Pr(D/\theta_{MC}, G) Pr(\theta_{MC}/G)$$

$$\log Pr(D/G) = \frac{d}{2}\log 2\pi - \frac{1}{2}\log|H| + \log\left(Pr(D/\theta_{MC}, G) Pr(\theta_{MC}/G)\right)$$

$$= O(1) - \frac{1}{2}\log|H| + \ell(\theta_{MC}; G)$$

$H = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix}_{d \times d}$

We can take $|H| \approx M^d \to$ get

$$\log(Pr(D/G)) = \ell(\theta_{MC}; G) - \frac{d}{2}\log M + O(1)$$

Bayesian Score

- As $M \to \infty$ Marginal Likelihood is equivalent to BIC score.
- That implies Bayesian score is equivalent to BIC score.
- The graph or its I-equivalent graph will have the highest score among all possible graphs.

A closer look at the formula:

$$\text{score}_{\text{BIC}}(G : D) = \ell(\hat{\theta} : G) - \frac{\log M}{2} \dim(G)$$

We can rewrite $\ell(\hat{\theta} : G)$ as follows

$$\ell(\hat{\theta} : G) = M \sum_{i=1}^{n} I_{\hat{P}}(X_i, Pa_{X_i}) - M \sum_{i=1}^{n} H_{\hat{P}}(X_i)$$

$$- \frac{\log M}{2} \dim(G)$$

$$score_{BIC}(G:D) = M \sum_{i=1}^{n} I_{\hat{P}}(X_i, Pa_{X_i}) - M \sum_{i=1}^{n} H_{\hat{P}}(X_i) - \frac{\log M}{2} \dim(G)$$

- Entropy terms do not depend on the graph and can be ignored
- The stronger the dependence between a variable and its parents, the higher the score
- The more complex the network, the lower the score
- Mutual information term grows linearly in M, complexity term grows logarithmically $\Rightarrow$ more the data, more emphasis given to fitting data

Table of Contents

Optimization Problem

- Input
  - Training data
  - Scoring function
  - Set of possible structures
- Output
  - Network that maximizes the score
- Key property for computation efficiency
  - Decomposibility of score

$$score(\mathcal{G} : \mathcal{D}) = \sum_i score(X_i \mid Pa_{X_i}^{\mathcal{G}} : \mathcal{D})$$

Learning Trees / Forests

$$score(\mathcal{G} : \mathcal{D}) = \sum_i score(X_i \mid Pa_{X_i}^{\mathcal{G}} : \mathcal{D})$$

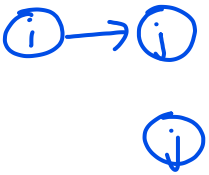
$$= \left[\sum_{i:p(i)>0} score(X_i \mid X_{p(i)} : \mathcal{D}) - score(X_i : \mathcal{D}) \right] + \sum_{i=1}^{n} score(X_i : \mathcal{D})$$

- Score = sum of edge scores + constant.

Algorithm for Learning Trees / Forests

Algorithm

- Define unirected graph with nodes $\{1, \ldots, n\}$
- Set $w(i \to j) = \max[score(X_j|X_i) - score(X_j), 0]$
- Find forest with maximal weights
  - Use MST algorthims like Prim's or Kruskal's
  - Remove edges with weight 0.



- Generates tree if likelihood score is used and a forest if BIC score is used.
- Network with only one parent.

General Network

- For general trees, with # parents ≥ 2, use
  - Heuristic hill climbing algorithm
  - Best first search
  - Simulated annealing

Greedy Hill Climbing Algorithm

- Start with a given network
  - Empty network
  - Best tree
  - Random network
  - Prior knowledge
- At each iteration
  - Consider score of all possible changes like addition, removal and reversal of edges.
  - Apply change that most improves the score (greedy approach)
- Stop when no modification improves score. This gives local maxima.

Greedy Hill Climbing Algorithm

- Pitfalls
  - Local maxima
  - Plateau – often in BN due to I-equivalent structures.
- To remove pitfalls
  - Use edge reversals
  - Random restarts
    - Take some random steps and then start climbing again
  - Tabu lists
    - Keep a list of most recent K steps and search cannot reverse any of these steps.

Chow Liu Algorithm

Chow-Liu Algorithm

Step 1: For each pair of variables A,B, use data to estimate P(A,B), P(A), P(B)

Step 2: For each pair of variables A,B, calculate mutual information

$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

Step 3: Calculate the maximum spanning tree over the set of variables, using edge weights I(A,B) (given N variables, this costs only $O(N^2)$ time)

Step 4: Add arrows to edges to form a directed-acyclic graph by picking an arbitrary node as root and directing edges outward from the root

Step 5: Learn the CPD's for this graph

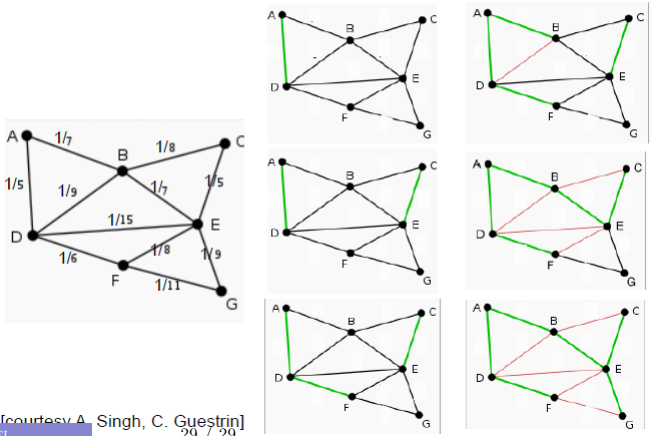
Kruskal's Algorithm

Maximum Spanning Tree Algorithm

- Kruskal's Algorithm
  - Start with the empty graph and add edges one by one
  - As the next edge to add, choose one that
    - Is not in graph yet
    - Does not introduce a cycle. Has the maximum weight

Chow Liu Algorithm

Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree



[courtesy A. Singh, C. Guestrin]