Scoring Functions for Learning Bayesian Networks

Brandon Malone

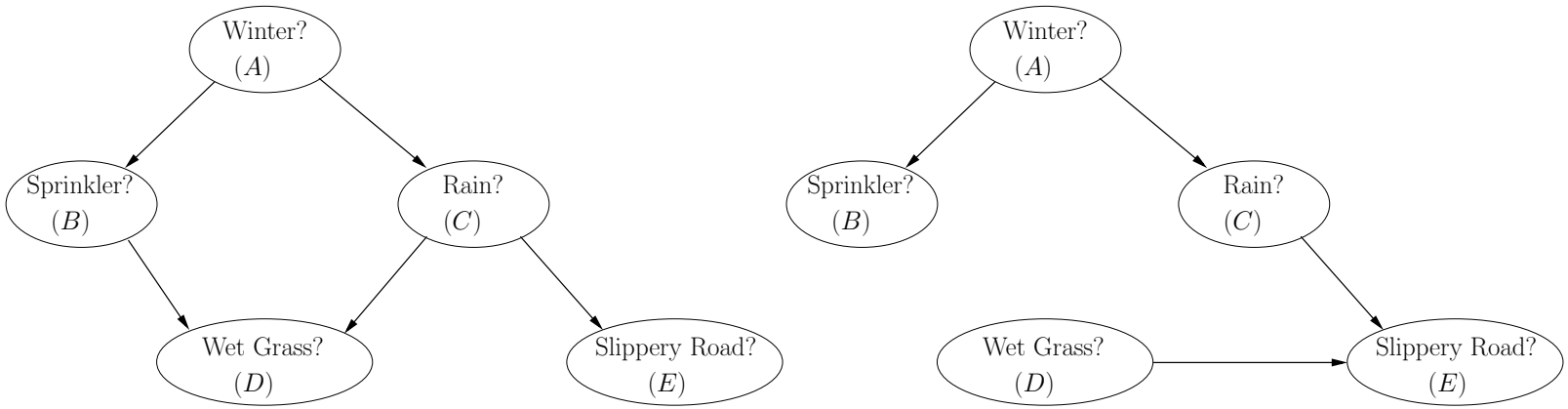
Much of this material is adapted from Suzuki 1993, Lam and Bacchus 1994, and Heckerman 1998

Many of the images were taken from the Internet

February 13, 2014

Scoring Functions for Learning Bayesian Networks

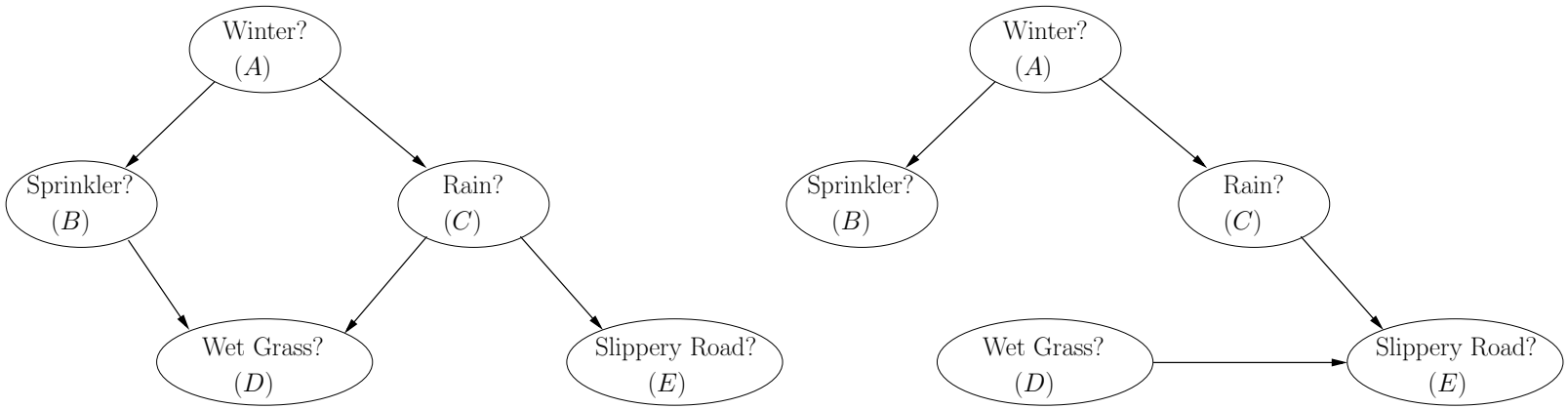
Suppose we have two Bayesian network structure $\mathcal{N}_1$ and $\mathcal{N}_2$.



Which structure best explains a dataset $\mathcal{D}$?

Scoring Functions for Learning Bayesian Networks

Suppose we have two Bayesian network structure $\mathcal{N}_1$ and $\mathcal{N}_2$.



Which structure best explains a dataset $\mathcal{D}$?

We will use **scoring functions** to rate each network.
The one with the best score is "better."

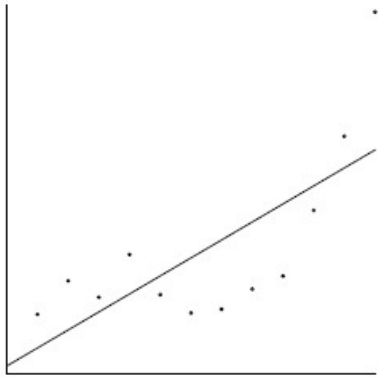
Why do we want to learn structures?

- Knowledge discovery ("interpretation")
- Density estimation ("prediction")

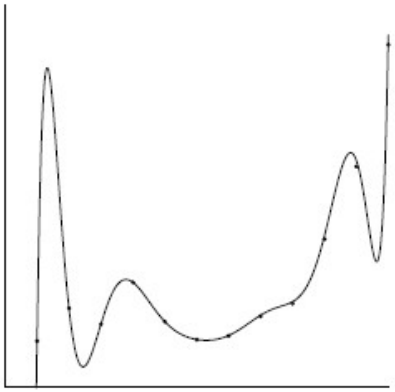
Assumptions (generally)

- Multinomial samples
- Complete data
- Parameter independence
  - Global
  - Local
- Parameter modularity

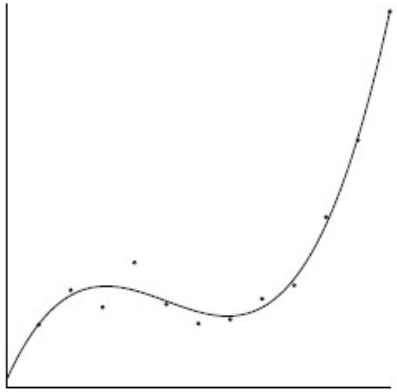
Under and overfitting



Underfitting, too simple



Overfitting, too complex



Tradeoff, "just right"

What does it mean in Bayesian networks?

Minimum description length (MDL)

MDL* views *learning as data compression*.

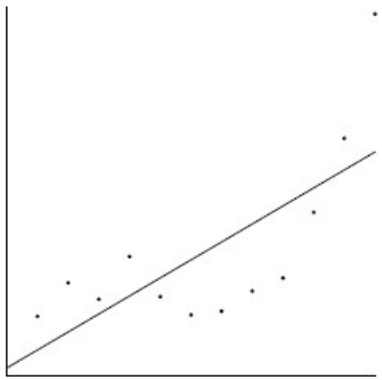
Traditionally, MDL consists of two components.

- Model encoding
- Data encoding, using the model

A few properties

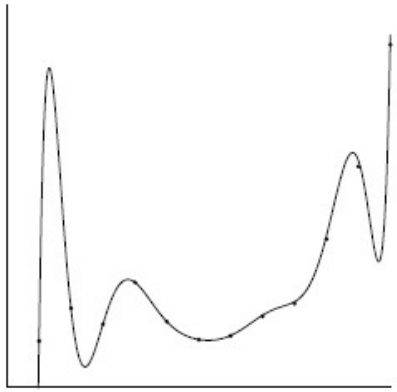
- Formalizes Occam's Razor
- Works regardless of a "true" model

Avoiding overfitting with MDL



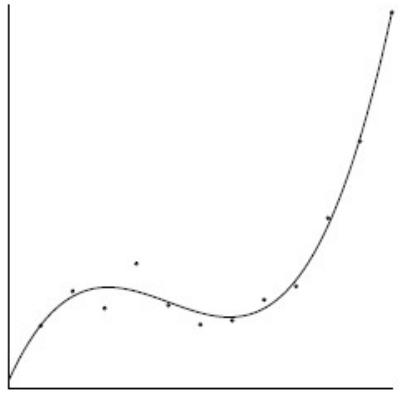
Short model encoding

Long data encoding



Long model encoding

Short data encoding



Medium model encoding

Medium data encoding

We will favor models which do not use too many bits to encode either the model or the data.

Encoding a Bayesian network

We must encode:

- Parents of each node

  We need $\log_2 n$ bits for each parent.

- Conditional probability parameters

  We need $(r_i - 1) \cdot q_i$ parameters for X_i.

  We need $\frac{\log_2 N}{2}$ bits per parameter.

The total complexity is as follows.

$$\sum_{i}^{n} \log n \cdot |PA_i| + \frac{\log N}{2} \cdot (r_i - 1) \cdot q_i$$

Other encodings are possible.

Encoding data with a Bayesian network

Each **complete instantiation** D_l is assigned a binary string (codeword) with length $\approx -\log p_l$.
We can approximate this value using the counts from the data.

$$
\begin{aligned}
p_l &= P(D_l|\mathcal{D},\mathcal{N}) \\
&= \prod_i^n \theta_{ijk:l} && \text{Chain rule of BNs} \\
&= \prod_i^n \frac{N_{ijk:l}}{N_{ij:l}} && \text{Using MLE parameters}
\end{aligned}
$$

Encoding data with a Bayesian network

Each **complete instantiation** D_l is assigned a binary string (codeword) with length $\approx -\log p_l$.

$$
\begin{aligned}
len(\mathcal{D} : \mathcal{N}) &= \sum_{l}^{N} len(D_l : \mathcal{N}) \\
&= \sum_{l}^{N} -\log \prod_{i}^{n} \frac{N_{ijk:l}}{N_{ij:l}} \\
&= -\sum_{l}^{N} \log \prod_{i}^{n} \prod_{k}^{r_i} \prod_{j}^{q_i} \frac{N_{ijk:l}}{N_{ij:l}} \\
&= -\sum_{l}^{N} \sum_{i}^{n} \sum_{j}^{q_i} \sum_{k}^{r_i} \log \frac{N_{ijk:l}}{N_{ij:l}} \\
&= -\sum_{i}^{n} \sum_{j}^{q_i} \sum_{k}^{r_i} N_{ijk} \cdot \log \frac{N_{ijk}}{N_{ij}}
\end{aligned}
$$

This is the log-likelihood, ℓ, of the data using the MLE parameters.

MDL as a scoring function

As derived here, the MDL score for a network $\mathcal{N}$ given a dataset $\mathcal{D}$ is as follws.

$$MDL(\mathcal{N}:\mathcal{D}) = -\sum_{i}^{n}\left\{\sum_{j}^{q_i}\sum_{k}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}}\right\} + \log n \cdot |PA_i| + \frac{\log N}{2}\cdot(r_i-1)\cdot q_i$$

As the dataset (N) grows, the $\log n \times |PA_i|$ term vanishes, so the most commonly used version of MDL is as follows.

$$MDL(\mathcal{N}:\mathcal{D}) = -\sum_{i}^{n}\left\{\sum_{j}^{q_i}\sum_{k}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}}\right\} + \frac{\log N}{2}\cdot(r_i-1)\cdot q_i$$

$$MDL(\mathcal{N}:\mathcal{D}) = -\sum_{i}^{n}\ell(X_i|PA_i) + \frac{\log N}{2}\cdot(r_i-1)\cdot q_i$$

Bayesian Dirichlet (BD) Score Family

Suppose we would like to maximize the joint probability of the data $\mathcal{D}$ and network $\mathcal{N}$.

$$\begin{aligned}\mathcal{N}^* &= \arg\max_{\mathcal{N}} P(\mathcal{D}, \mathcal{N}) \\ &= \arg\max_{\mathcal{N}} P(\mathcal{D}|\mathcal{N})P(\mathcal{N})\end{aligned}$$

We again have two parts.

- Evaluation of the model, $P(\mathcal{N})$
- Evaluation of data given the model, $P(\mathcal{D}|\mathcal{N})$

Data given the model

We need to evaluate $P(\mathcal{D}|\mathcal{N})$. We derived $P(\mathbf{x}_l|\mathcal{D},\mathcal{N})$ for parameter estimation.

$$P(\mathbf{x}_l|\mathcal{D},\mathcal{N}) = \prod_i^n \prod_j^{q_i} \prod_k^{r_i} \frac{\alpha_{ijk:l} + n_{ijk:l}}{\sum_k (\alpha_{ijk:l} + n_{ijk:l})}$$

Because the samples are iid, we can evaluate $P(\mathcal{D}|\mathcal{N})$ by taking the product.

$$\begin{aligned} P(\mathcal{D}|\mathcal{N}) &= \prod_l^N \prod_i^n \prod_j^{q_i} \prod_k^{r_i} \frac{\alpha_{ijk:l} + n_{ijk:l}}{\sum_k (\alpha_{ijk:l} + n_{ijk:l})} \\ &= \prod_i^n \prod_j^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_k^{r_i} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

Score probability, $P(\mathcal{D}, \mathcal{N})$

We are interested in the joint probability of $\mathcal{D}$ and $\mathcal{N}$.

$$P(\mathcal{D}, \mathcal{N}) = P(\mathcal{N})P(\mathcal{D}|\mathcal{N})$$
$$= P(\mathcal{N}) \prod_{i}^{n} \prod_{j}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k}^{r_i} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}$$

This is called the BD scoring function.

If we set $\alpha_{ijk} = 1$, then it is called the $K2$ metric.

Some desirable equivalences

Say $\mathcal{N}_1$and $\mathcal{N}_2$ are Markov equivalent.

- Prior probabilities and equivalence

$$P(\mathcal{N}_1) = P(\mathcal{N}_2)$$

- Likelihood probabilities and equivalence

$$P(\mathcal{D}|\mathcal{N}_1) = P(\mathcal{D}|\mathcal{N}_2)$$

- Score probabilities and equivalence

$$P(\mathcal{D}, \mathcal{N}_1) = P(\mathcal{D}, \mathcal{N}_2)$$

Some desirable equivalences

Say $\mathcal{N}_1$and $\mathcal{N}_2$ are Markov equivalent.

- Prior probabilities and equivalence

$$P(\mathcal{N}_1) = P(\mathcal{N}_2)$$

- Likelihood probabilities and equivalence **Not guaranteed by BD**

$$P(\mathcal{D}|\mathcal{N}_1) = P(\mathcal{D}|\mathcal{N}_2)$$

- Score probabilities and equivalence

$$P(\mathcal{D}, \mathcal{N}_1) = P(\mathcal{D}, \mathcal{N}_2)$$

BDe and BDeu

We can restrict the hyperparameters to ensure likelihood **e**quivalence. This is **BDe**.

$$\alpha_{ijk} = \alpha \cdot P(X_i = k, PA_i = j | \mathcal{N})$$

Typically, **u**ninformative hyperparameters are used. This is **BDeu**.

$$\alpha_{ijk} = \frac{\alpha}{r_i \cdot q_i}$$

The BDeu scoring function

We can incorporate our assumptions to derive the BDeu scoring function.

$$P(\mathcal{D},\mathcal{N}) = P(\mathcal{N})P(\mathcal{D}|\mathcal{N}) \quad \text{Rewrite using chain rule}$$

$$= P(\mathcal{N})\prod_i^n\prod_j^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+n_{ij})}\prod_k^{r_i}\frac{\Gamma(\alpha_{ijk}+n_{ijk})}{\Gamma(\alpha_{ijk})} \quad \text{Substitute probability of data}$$

$$\propto \prod_i^n\prod_j^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+n_{ij})}\prod_k^{r_i}\frac{\Gamma(\alpha_{ijk}+n_{ijk})}{\Gamma(\alpha_{ijk})} \quad \text{Assume a uniform structure prior}$$

$$\propto \prod_i^n\prod_j^{q_i}\frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(\frac{\alpha}{q_i}+n_{ij})}\prod_k^{r_i}\frac{\Gamma(\frac{\alpha}{r_i\cdot q_i}+n_{ijk})}{\Gamma(\frac{\alpha}{r_i\cdot q_i})} \quad \text{Replace the } \alpha\text{s}$$

$$BDeu(\mathcal{N}:\mathcal{D},\alpha) = \sum_i^n\sum_j^{q_i}\log\frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(\frac{\alpha}{q_i}+n_{ij})} + \sum_k^{r_i}\log\frac{\Gamma(\frac{\alpha}{r_i\cdot q_i}+n_{ijk})}{\Gamma(\frac{\alpha}{r_i\cdot q_i})} \quad \text{Work in log-space}$$

$$BDeu(\mathcal{N}:\mathcal{D},\alpha) = \sum_i^n\sum_j^{q_i}\log\Gamma(\frac{\alpha}{q_i}) - \log\Gamma(\frac{\alpha}{q_i}+n_{ij}) + \quad \text{Remove divisions}$$

$$\sum_k^{r_i}\log\Gamma(\frac{\alpha}{r_i\cdot q_i}+n_{ijk}) - \log\Gamma(\frac{\alpha}{r_i\cdot q_i})$$

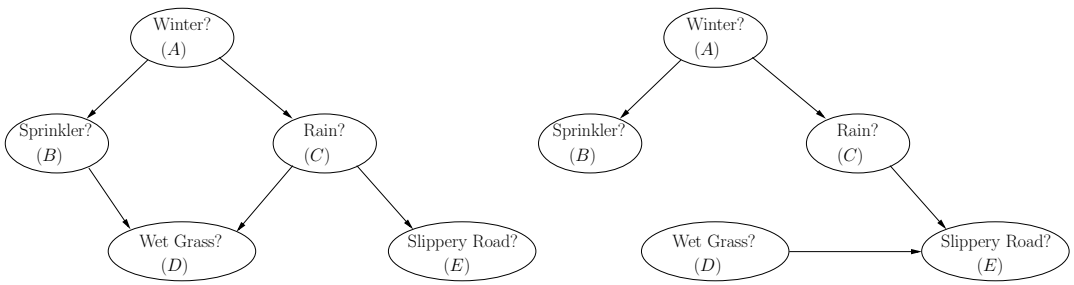
Decomposability

Both MDL and BD are **decomposable**: a sum over terms which involve only a variable and its parents.

$$MDL(\mathcal{N}:\mathcal{D}) = -\sum_{i}^{n}\left\{\ell(X_i|PA_i) + \frac{\log N}{2}\cdot(r_i-1)\cdot q_i\right\}$$

$$BDeu(\mathcal{N}:\mathcal{D},\alpha) = \sum_{i}^{n}\left\{\sum_{j}^{q_i}\log\Gamma(\frac{\alpha}{q_i}) - \log\Gamma(\frac{\alpha}{q_i}+n_{ij}) + \sum_{k}^{r_i}\log\Gamma(\frac{\alpha}{r_i\cdot q_i}+n_{ijk}) - \log\Gamma(\frac{\alpha}{r_i\cdot q_i})\right\}$$

What does it mean when we evaluate different structures?



Limitations of scoring functions

- Parameter independence is violated if data is missing.
- Experimental data is different that observational data.
- (MDL) When do we use asymptotics?
- (BD) How do we specify α and $P(\mathcal{N})$?

Recap

During this part of the course, we have discussed:

- Overfitting
- Minimum description length scoring function for BNs
- BD family of scores for BNs

Next in probabilistic models

We will discuss two strategies for learning Bayesian network structures.

- A greedy hill climbing algorithm which finds local optima
- A dynamic programming algorithm which guarantees to find an optimal network