Sequential and Factorized NML models

Tomi Silander  Teemu Roos  Petri Myllymäki

Helsinki Institute for Information Technology HIIT

Recent Breakthroughs in Minimum Description Length Learning,
ICML/UAI/COLT Workshop, 2008
1. Task: Learning Bayesian Network from the Data
   - Data
   - Bayesian networks
   - Learning the Structure
   - Learning the Parameters

2. Current State of the Art
   - Bayesian Mixture
   - Expected Parameter Values
   - Problem: Sensitivity to the Prior

3. An Efficient NML-based Approach
   - Factorized NML
   - Sequential NML Parameter Values
Let
\[ x^n := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix} = \begin{pmatrix} x_{1,:} \\ x_{2,:) \\ \vdots \\ x_{n,:} \end{pmatrix} = (x_{:,1} x_{:,2} \cdots x_{:,m}) \]

be a data matrix where each row, \( x_{i,:} = (x_{i,1}, x_{i,2}, \ldots, x_{i,m}), 1 \leq i \leq n \), is an \( m \)-dimensional observation vector, and columns of \( x^n \) are denoted by \( x_{:,1}, \ldots, x_{:,m} \).

- The multidimensional rows \( x_{i,:) \) are assumed i.i.d.
- There can be dependencies between the dimensions (columns \( x_{:,1}, \ldots, x_{:,m} \)).
Let $x^n := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix} = \begin{pmatrix} x_{1,:} \\ x_{2,:} \\ \vdots \\ x_{n,:) \end{pmatrix} = (x_{:,1}x_{:,2}\cdots x_{:,m})$

be a data matrix where each row, $x_{i,:} = (x_{i,1}, x_{i,2}, \ldots, x_{i,m}), 1 \leq i \leq n$, is an $m$-dimensional observation vector, and columns of $x^n$ are denoted by $x_{:,1}, \ldots, x_{:,m}$.

- The multidimensional rows $x_{i,:}$ are assumed i.i.d.
- There can be dependencies between the dimensions (columns $x_{:,1}, \ldots, x_{:,m}$).
Let

\[ x^n := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix} = \begin{pmatrix} x_{1,:} \\ x_{2,:} \\ \vdots \\ x_{n,:) \end{pmatrix} = (x_{:,1} x_{:,2} \cdots x_{:,m}) \]

be a data matrix where each row, \( x_{i,:} = (x_{i,1}, x_{i,2}, \ldots, x_{i,m}), 1 \leq i \leq n \), is an \( m \)-dimensional observation vector, and columns of \( x^n \) are denoted by \( x_{:,1}, \ldots, x_{:,m} \).

The multidimensional rows \( x_{i,:} \) are assumed i.i.d.

There can be dependencies between the dimensions (columns \( x_{:,1}, \ldots, x_{:,m} \)).
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.
- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:
  \[ p(x^n; \theta) = \prod_{j=1}^{m} p(x_{i,j} | Pa_j; \theta_{j|Pa_j}) \]
- E.g., given the network above, we get:
  \[ P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7). \]
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.

- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:

\[
p(x^n; \theta) = \prod_{j=1}^m p(x_{i,j} | Pa_j ; \theta|_{Pa_j})
\]

- E.g., given the network above, we get

\[
P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7).
\]
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.
- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:

\[
p(x^n ; \theta) = \prod_{j=1}^{m} p(x_{:,j} | Pa_j ; \theta_{j|Pa_j})
\]

- E.g., given the network above, we get:

\[
P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = 
P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7).
\]
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.

- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:

  \[ p(x^n ; \theta) = \prod_{j=1}^m p(x_{.,j} | Pa_j ; \theta_j|Pa_j) \]

- E.g., given the network above, we get

  \[
P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = \]
  \[
P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7).
\]
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.

- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:

\[ p(x^n ; \theta) = \prod_{j=1}^{m} p(x_{:.j} | Pa_j ; \theta_{j|Pa_j}) \]

- E.g., given the network above, we get

\[
P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = 
P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7).
\]
Bayesian networks

- In general, a Bayesian network is a DAG representing a set of independence assumptions.

- In particular, given a Bayesian network, the joint distribution factorizes as a product of local distributions, each conditioned on the parents of a node:

\[
p(x^n ; \theta) = \prod_{j=1}^{m} p(x_{:,j} \mid Pa_j ; \theta_j|Pa_j)
\]

- E.g., given the network above, we get

\[
P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_2, X_4)P(X_6)P(X_7|X_6)P(X_8|X_7).
\]
Learning the best Bayesian network structure is NP hard for all common learning criteria such as AIC, BIC, and Bayesian Mixture.

For NML it is super inefficient to score even a single network.

The score based learning heuristics are based on the decomposability of the learning criterion

\[
SCORE(G, x^n) = \sum_{j=1}^{m} Score(x_{-j}, x_{-j}, Pa_j).
\]

This allows incremental evaluation of the network score in local search.
Learning the best Bayesian network structure is NP hard for all common learning criteria such as AIC, BIC, and Bayesian Mixture.

For NML it is super inefficient to score even a single network.

The score based learning heuristics are based on the decomposability of the learning criterion

$$
SCORE(G, x^n) = \sum_{j=1}^{m} \text{Score}(x_{j-1}, x_j, Pa_j).
$$

This allows incremental evaluation of the network score in local search.
Learning the best Bayesian network structure is NP hard for all common learning criteria such as AIC, BIC, and Bayesian Mixture. For NML it is super inefficient to score even a single network. The score based learning heuristics are based on the decomposability of the learning criterion

\[ \text{SCORE}(G, x^n) = \sum_{j=1}^{m} \text{Score}(x_{:,j}, x_{:, Pa_j}). \]

This allows incremental evaluation of the network score in local search.
Learning the best Bayesian network structure is NP hard for all common learning criteria such as AIC, BIC, and Bayesian Mixture.

For NML it is super inefficient to score even a single network.

The score based learning heuristics are based on the decomposability of the learning criterion

$$SCORE(G, x^n) = \sum_{j=1}^{m} \text{Score}(x_{-j}, x_{-j}, Pa_j).$$

This allows incremental evaluation of the network score in local search.
Bayesian networks are often used for predictive purposes. Therefore it is important to find good parameters for the learned structure. Some learning criteria like Bayesian mixture, naturally suggest a parameterization while others (AIC, BIC) do not. Maximum likelihood parameters are usually a very bad choice, due to zero frequencies in multivariate discrete data.
Learning the Parameters

Bayesian networks are often used for predictive purposes.

Therefore it is important to find good parameters for the learned structure.

Some learning criteria like Bayesian mixture, naturally suggest a parameterization while others (AIC, BIC) do not.

Maximum likelihood parameters are usually a very bad choice, due to zero frequencies in multivariate discrete data.
Bayesian networks are often used for predictive purposes.
Therefore it is important to find good parameters for the learned structure.
Some learning criteria like Bayesian mixture, naturally suggest a parameterization while others (AIC, BIC) do not.
Maximum likelihood parameters are usually a very bad choice, due to zero frequencies in multivariate discrete data.
Learning the Parameters

- Bayesian networks are often used for predictive purposes.
- Therefore it is important to find good parameters for the learned structure.
- Some learning criteria like Bayesian mixture, naturally suggest a parameterization while others (AIC, BIC) do not.
- Maximum likelihood parameters are usually a very bad choice, due to zero frequencies in multivariate discrete data.
Bayesian Mixture

- Bayesian criterion would be to select the network structure $\mathcal{G}$ by its posterior probability

$$P(\mathcal{G} \mid x^n, \alpha) \propto P(x^n \mid \mathcal{G}, \alpha)P(\mathcal{G})$$

$$= P(\mathcal{G}) \int_{\Theta} P(x^n \mid \Theta, \mathcal{G})P(\Theta \mid \mathcal{G}, \alpha)d\Theta.$$

- With comfortable choice of parameter independence assumptions, decomposable structure prior, and conjugate parameter priors, the criterion is decomposable and it has a closed form solution.

- With uniform prior for structures, this coincides with a prequential selection criterion

$$P(x^n \mid \mathcal{G}, \alpha) = \prod_{i=1}^{n} P(x_i, : \mid x^{i-1}, \mathcal{G}, \alpha),$$

that does not depend on the order of rows.
Bayesian criterion would be to select the network structure $\mathcal{G}$ by its posterior probability

$$P(\mathcal{G} \mid x^n, \alpha) \propto P(x^n \mid \mathcal{G}, \alpha) P(\mathcal{G})$$

$$= P(\mathcal{G}) \int_{\Theta} P(x^n \mid \Theta, \mathcal{G}) P(\Theta \mid \mathcal{G}, \alpha) d\Theta.$$

With comfortable choice of parameter independence assumptions, decomposable structure prior, and conjugate parameter priors, the criterion is decomposable and it has a closed form solution.

With uniform prior for structures, this coincides with a prequential selection criterion

$$P(x^n \mid \mathcal{G}, \alpha) = \prod_{i=1}^{n} P(x_i \mid x^{i-1}, \mathcal{G}, \alpha),$$

that does not depend on the order of rows.
Bayesian Mixture

- Bayesian criterion would be to select the network structure $\mathcal{G}$ by its posterior probability

$$P(\mathcal{G} \mid x^n, \alpha) \propto P(x^n \mid \mathcal{G}, \alpha)P(\mathcal{G})$$

$$= P(\mathcal{G}) \int_{\Theta} P(x^n \mid \Theta, \mathcal{G})P(\Theta \mid \mathcal{G}, \alpha)d\Theta.$$  

- With comfortable choice of parameter independence assumptions, decomposable structure prior, and conjugate parameter priors, the criterion is decomposable and it has a closed form solution.

- With uniform prior for structures, this coincides with a prequential selection criterion

$$P(x^n \mid \mathcal{G}, \alpha) = \prod_{i=1}^{n} P(x_i, \cdot \mid x_i^{-1}, \mathcal{G}, \alpha),$$

that does not depend on the order of rows.
A common choice is to select the parameters $\tilde{\theta}$ so that they mimic the process used in sequential prediction:

$$P(x_{n+1,:} \mid \tilde{\theta}, G) = P(x_{n+1,:} \mid x^n, G, \alpha)$$

$$= \int_{\Theta} P(x_{n+1,:} \mid \Theta, G)P(\Theta \mid x^n, G, \alpha)d\Theta$$

Under the assumptions made already for structure learning, the $\tilde{\theta}$ equals the expected parameter values $\tilde{\theta} = E[\Theta \mid x^n, G, \alpha]$. 
A common choice is to select the parameters $\tilde{\theta}$ so that they mimic the process used in sequential prediction

$$P(x_{n+1},: | \tilde{\theta}, G) = P(x_{n+1},: | x^n, G, \alpha)$$

$$= \int_\Theta P(x_{n+1},: | \Theta, G) P(\Theta | x^n, G, \alpha) d\Theta$$

Under the assumptions made already for structure learning, the $\tilde{\theta}$ equals the expected parameter values $\tilde{\theta} = E[\Theta | x^n, G, \alpha]$. 

Expected parameter values
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | \mathcal{G}, \alpha)$.

- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.

- Posterior odds of the models are very sensitive to the prior too.

- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - More data: less arcs
    - Even more data: many arcs again.
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | \mathcal{G}, \alpha)$.

- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.

- Posterior odds of the models are very sensitive to the prior too.

- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - More data: less arcs
    - Even more data: many arcs again.
The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | \mathcal{G}, \alpha)$.

This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.

Posterior odds of the models are very sensitive to the prior too.

Other oddities as well:
- The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
- Complexity oscillates:
  - Little data: many arcs
  - More data: less arcs
  - Even more data: many arcs again.
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | \mathcal{G}, \alpha)$.
- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.
- Posterior odds of the models are very sensitive to the prior too.
- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - more data: less arcs;
    - even more data: many arcs again.
The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | \mathcal{G}, \alpha)$. This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.

Posterior odds of the models are very sensitive to the prior too.

Other oddities as well:

- The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.

Complexity oscillates:
- Little data: many arcs
- More data: less arcs;
- Even more data: many arcs again.
Problem: Sensitivity to the Prior

The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta | G, \alpha)$.

This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.

Posterior odds of the models are very sensitive to the prior too.

Other oddities as well:

- The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
- Complexity oscillates:
  - Little data: many arcs
  - more data: less arcs;
  - even more data: many arcs again.
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta \mid \mathcal{G}, \alpha)$.
- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.
- Posterior odds of the models are very sensitive to the prior too.
- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - more data: less arcs;
    - even more data: many arcs again.
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta \mid \mathcal{G}, \alpha)$.
- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.
- Posterior odds of the models are very sensitive to the prior too.
- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - more data: less arcs;
    - even more data: many arcs again.
Problem: Sensitivity to the Prior

- The Bayesian model selection criterion is very sensitive to the choice of the prior $P(\Theta \mid \mathcal{G}, \alpha)$.
- This is mainly due to the nestedness of the models. Choice of priors determines which model in nested hierarchy equipped with its expected parameters best approximates the (joint) probability of a data row.
- Posterior odds of the models are very sensitive to the prior too.
- Other oddities as well:
  - The prior promoting uniform probability of data vectors may cause two skewed independent variables appear dependent.
  - Complexity oscillates:
    - Little data: many arcs
    - more data: less arcs;
    - even more data: many arcs again.
Factorized NML

- The NML denominator is way too costly to compute.
- But the decomposable variant can be computed

\[ P_{fNML}(x^n) = \prod_{j=1}^{m} P_{NML}(x_{:,j} \mid x_{:,Pa_j}, G), \]

where

\[ P_{NML}(x_{:,j} \mid x_{:,Pa_j}, G) = \frac{P(x_{:,j} \mid \hat{\theta}(x_{:,j}, x_{:,Pa_j}))}{\sum_{x'_{:,j}} P(x'_{:,j} \mid \hat{\theta}(x'_{:,j}, x_{:,Pa_j}))}. \]

- The normalizer further decomposes into multinomial regrets, which makes it as fast to compute than other popular scores for learning Bayesian network structures.
Factorized NML

- The NML denominator is way too costly to compute.
- But the decomposable variant can be computed

\[
P_{fNML}(x^n) = \prod_{j=1}^{m} P_{NML}(x_{.:j} | x_{.:Pa_j}, G),
\]

where

\[
P_{NML}(x_{.:j} | x_{.:Pa_j}, G) = \frac{P(x_{.:j} | \hat{\theta}(x_{.:j}, x_{.:Pa_j}))}{\sum_{x'_{.:j}} P(x'_{.:j} | \hat{\theta}(x'_{.:j}, x_{.:Pa_j}))}.
\]

- The normalizer further decomposes into multinomial regrets, which makes it as fast to compute than other popular scores for learning Bayesian network structures.
Factorized NML

- The NML denominator is way too costly to compute.
- But the decomposable variant can be computed

\[
P_{fNML}(x^n) = \prod_{j=1}^{m} P_{NML}(x_{:,j} | x_{:,Pa_j}, G),
\]

where

\[
P_{NML}(x_{:,j} | x_{:,Pa_j}, G) = \frac{P(x_{:,j} \mid \hat{\theta}(x_{:,j}, x_{:,Pa_j}))}{\sum_{x_{:,j}'} P(x_{:,j}' \mid \hat{\theta}(x_{:,j}', x_{:,Pa_j}))}.
\]

- The normalizer further decomposes into multinomial regrets, which makes it as fast to compute than other popular scores for learning Bayesian network structures.
Sequential NML Parameter Values

- The natural parameterization for the structure selected by fNML is obtained by

\[ P(x_{n+1}, \cdot \mid x^n, \mathcal{G}) \propto P_{fNML}(x^{n+1}) \]

\[ = \prod_{j=1}^{m} P_{NML}(x_{n+1,j} \mid x_{n+1}, \text{pa}_j, \mathcal{G}) \]

\[ = \prod_{j=1}^{m} \frac{P(x_{n+1,j} \mid \hat{\theta}(x_{n+1,j}, x_{n+1}, \text{pa}_j))}{\sum_{x'_{n+1,j}} P(x'_{n+1,j} \mid \hat{\theta}(x'_{n+1,j}, x'_{n+1}, \text{pa}_j))}. \]

This distribution can be obtained by setting the Bayesian network parameters

\[ \theta_{j\mid \text{pa}_j}(r, s) = \frac{e(N_{jsr})(N_{jsr} + 1)}{\sum_{r'} e(N_{jsr'})(N_{jsr'} + 1)}, \]

where \( e(n) = \left(\frac{n+1}{n}\right)^n \) and \( N_{jsr} = \left| \{i : x_{i,j} = r, \text{pa}_i,j = s\} \right| \).
Sequential NML Parameter Values

- The natural parameterization for the structure selected by fNML is obtained by

\[ P(x_{n+1} : | x^n, G) \propto P_{fNML}(x^{n+1}) \]

\[ = \prod_{j=1}^{m} P_{\text{NML}}(x_{n+1,j} : | x_{n+1}, Pa_j, G) \]

\[ = \prod_{j=1}^{m} \frac{P(x_{n+1,j} : | \hat{\theta}(x_{n+1,j}, x_{n+1}, Pa_j))}{\sum_{x'_{n+1,j}} P(x'_{n+1,j} : | \hat{\theta}(x'_{n+1,j}, x'_{n+1}, Pa_j))}. \]

- This distribution can be obtained by setting the Bayesian network parameters

\[ \theta_{j|Pa_j}(r, s) = \frac{e(N_{jsr})(N_{jsr} + 1)}{\sum_{r'} e(N_{jsr'})(N_{jsr'} + 1)}, \]

where \( e(n) = \left( \frac{n+1}{n} \right)^n \) and \( N_{jsr} = |\{i : x_{i,j} = r, pa_{i,j} = s\}|. \)
Learning Bayesian networks is hard

- The current state-of-the-art uses the Bayesian mixture as a model selection criterion but it has prior sensitivity problems.
- Using factorized NML for learning the structure, and sequential NML for learning the parameters provides an efficient and care-free alternative.
Summary

- Learning Bayesian networks is hard
- The current state-of-the-art uses the Bayesian mixture as a model selection criterion but it has prior sensitivity problems.
- Using factorized NML for learning the structure, and sequential NML for learning the parameters provides an efficient and care-free alternative.
Learning Bayesian networks is hard

The current state-of-the-art uses the Bayesian mixture as a model selection criterion but it has prior sensitivity problems.

Using factorized NML for learning the structure, and sequential NML for learning the parameters provides an efficient and care-free alternative.