Information theoretic model selection criteria in score-based Bayesian network structure learning

Tomi Silander
NAVER LABS Europe
Grenoble, France

May 28, 2018

Outline

- Preliminaries
- Motivation
- Information theoretic criteria
- Conclusion

NAVER LABS Europe (Grenoble)

- about 100 researchers (and growing) from 25 different countries
- machine learning and AI
  - graphical models, deep neural networks, optimization, reinforcement learning, ...
  - computer vision, natural language processing, robotics, mobility, ...

Outline

- Preliminaries
- Motivation
- Information theoretic criteria
- Conclusion
Super-exponentially many models

\[ B_n = \begin{cases} 
1 & \text{if } n = 0 \\
\sum_{k=1}^{n} (-1)^{k+1} \left( \begin{array}{c} n \\ k \end{array} \right) 2^{k(n-k)}B_{n-k} & \text{if } n > 0
\end{cases} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( B_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>0.333</td>
</tr>
<tr>
<td>3</td>
<td>0.040</td>
</tr>
<tr>
<td>4</td>
<td>0.00714</td>
</tr>
<tr>
<td>5</td>
<td>0.000344</td>
</tr>
<tr>
<td>6</td>
<td>0.0000264</td>
</tr>
<tr>
<td>7</td>
<td>0.000000000000000979</td>
</tr>
<tr>
<td>8</td>
<td>0.0000000000000000138</td>
</tr>
<tr>
<td>9</td>
<td>0.00000000000000000234</td>
</tr>
<tr>
<td>10</td>
<td>0.000000000000000000024</td>
</tr>
<tr>
<td>11</td>
<td>0.00000000000000000000316</td>
</tr>
<tr>
<td>12</td>
<td>0.00000000000000000000000152</td>
</tr>
<tr>
<td>13</td>
<td>0.000000000000000000000000535</td>
</tr>
<tr>
<td>14</td>
<td>0.0000000000000000000000000065</td>
</tr>
<tr>
<td>15</td>
<td>0.000000000000000000000000000421</td>
</tr>
<tr>
<td>16</td>
<td>0.0000000000000000000000000000019</td>
</tr>
<tr>
<td>17</td>
<td>0.000000000000000000000000000000101</td>
</tr>
<tr>
<td>18</td>
<td>0.0000000000000000000000000000000001</td>
</tr>
<tr>
<td>19</td>
<td>0.000000000000000000000000000000000001</td>
</tr>
<tr>
<td>20</td>
<td>0.0000000000000000000000000000000000001</td>
</tr>
</tbody>
</table>

Why select?

- Bayesians should not
  - neither for science nor for engineering
  - Especially since the most probable model has the probability 0.0000001.
    - so you “know” is wrong and select anyway?
  - But what about “description” or compression
  - gonna send all the probabilities of all the models?
    - Interpretability:
      - but think Berman’s Statistical Modeling: The Two Cultures
  - Causality
    - cognitive arguments

Model selection for what?

- **Scientific criterion** reveals the data generating mechanism: \( P(S|D; \alpha) \)
  - \( S \) = structure
  - \( D \) = data
  - \( \alpha \) = hyperparameters
- **Engineering criterion** finds the model that predicts well
  - \( P(d|D, \hat{S}; \alpha) \)
  - or even \( P(d|D; \alpha) = \sum S P(d|D, S; \alpha)P(S|D; \alpha). \)

Score-based vs. independence tests

*It is often stated* in papers tackling the task of inferring Bayesian network structures from data *that there are these two distinct approaches*:

- **Apply conditional independence tests when testing for the presence or otherwise of edges;**
- **Search the model space using a scoring metric.**

*Here I argue that* for complete data and a given node ordering *this division is a myth*, by showing that cross entropy methods for checking conditional independence are mathematically identical to methods based upon discriminating between models by their overall goodness-of-fit logarithmic scores.

R. G. Cowell. Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models. UAI 2001
Outline

- Preliminaries
- Motivation
- Information theoretic criteria
- Conclusion

Here’s my very practical problem

- Social scientists often have small sample sizes
  - Say ~60
- And when the ask me to learn a Bayesian network structure, I give them this:

![Bayesian Network Diagram]

\[
P(D|S; \alpha) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma\left(\sum_{k=1}^{r_{h}} \alpha_{ijk} + n_{ijk}\right)}{\Gamma\left(\sum_{k=1}^{r_{h}} \alpha_{ijk}\right) \Gamma\left(n_{ijk}\right)},
\]

where

- \( n \) is number of categorical variables \( X_1, \ldots, X_n \)
- \( r_i \) is the number of categorical values for variable \( X_i \)
- \( q_i \) is the number of value configurations of the parents \( Pa_i \) of \( i \) in the network \( S \).
  - \( q_i = \prod_{X_h \in Pa_i} r_h \), or \( q_i = 1 \), if \( Pa_i = \emptyset \).
- \( n_{ijk} \) is the number of times the variable \( X_i \) has a value \( k \) when its parents have the \( j^th \) value configuration in a data \( D \).
- \( X_i|Pa_i = j \sim \text{Cat}(\theta_{ij}) \), \( \theta_{ij} \sim \text{Dir}(\alpha_{ij1}, \ldots, \alpha_{ijq_i}) \).
Parameter sensitivity of Bayesian Dirichlet score

\[ P(D|S; \alpha) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r} \alpha_{ijk})}{\Gamma(\sum_{k=1}^{r} \alpha_{ijk} + n_{ijk})} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}. \]

- **BDeu**
  - \( \alpha_{ijk} = \frac{\alpha}{q_i} \)
  - is known to be very sensitive to a single Dirichlet hyperparameter \( \alpha \).
  - And it is not “regular” (Suzuki!!)
- Some proposed alternatives with parameters give up likelihood equivalence (one of the main motivations for BDeu)
  - But be Bayesian \( \alpha_{ijk} = \alpha \)
  - or like Bayesian Dirichlet sparse BDs(\( \alpha \)) a la \( \alpha_{ijk} = \frac{\alpha}{q_i} \), where \( q_i \)is the number of parent configurations of \( Pa_i \) that actually occur in the data \( D \) at least once (Scutari).
  - empirically Bayes

Parameter sensitivity remedies

- Fixing \( \alpha_{ijk} \) to some constant (like 1.0 or \( \frac{1}{2} \)) is one way to solve sensitivity.
- Using non-uniform structure prior can also help (?)
  - MU prior (marginal uniform): let it be a priori twice as probable to not have an arc than to have one (Scutari)
- Using criteria without parameters is an option
  - factorized Normalized Maximum Likelihood (fNML)
  - quotient Normalized Maximum Likelihood (qNML)
  - (discussed later)

But for now, let us look at the parameter sensitivity

**BDeu posterior at \( 0.1 \leq \alpha \leq 20.00 \)**

iris \( P(S|D; \alpha) \) BDeu

BDeu animation
Dir MAP predictive distributions $0.1 \leq \alpha \leq 20.00$

Dir predictive distributions $0.1 \leq \alpha \leq 20.00$

Dir predictive distributions $0.1 \leq \alpha \leq 20.00$ with MU

BDs posterior at $0.1 \leq \alpha \leq 20.00$
BDs predictive distributions $0.1 \leq \alpha \leq 20.00$ with MU

Comparing $P(S|D; \alpha)$ sensitivity

Comparing $P(d|D, \hat{S}; \alpha)$ sensitivity
Super exponentially many models

\[ B_n = \begin{cases} 
1 & \text{if } n = 0 \\
\sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} B_{n-k} & \text{if } n > 0 
\end{cases} \]

Often \( P(\hat{S}|D) \ll 1 \) (how can you tell)

- Calculate the not normalized probabilities \( cP(T_i) \) of likely structures \( T = \{ T_i \} \), other than MAP structure \( \hat{S} \).
  - \( cP(T) = \sum_i cP(T_i|D) \).
- Now even if all the other networks, i.e., those not in \( T \cup \{ \hat{S} \} \) had zero probability
  \[ P(\hat{S}|D) \leq \frac{cP(\hat{S}|D)}{cP(\hat{S}|D) + cP(T|D)}. \]
  - Often that alone makes \( P(\hat{S}|D) \ll 1 \)
  - People tend to get discouraged at \( P(\hat{S}|D) \approx 1\% \).

Sometimes with big data \( P(\hat{S}|D) \approx 1 \) (show that)

- Calculate the not normalized probability \( cP(\hat{S}|D) \) and the not normalized probability \( cP(S_2|D) \) of the second most probable network \( S_2 \) to get the ratio \( r = \frac{P(S_2|D)}{P(\hat{S}|D)} \).
- Let \( B \) be the number of all the possible nets.
- Now
  \[ P(\hat{S}|D) = \frac{P(\hat{S}|D)}{1 + \left( B - 1 \right)} P(S_2|D) \]
  \[ = \frac{1}{1 + (B - 1)r}. \]
  - Sometimes \( r \) is small that \( P(\hat{S}|D) \) is close to 1.0.
  - Adjust formula a bit for equivalent networks

Averaging over sub structures - not possible in general, but

- For small number on variables, one can learn the most probable order with dynamic programming and discrete Moebius transform.
  - Let \( G \preceq G^* \) mean that \( G \) is a subnetwork of \( G^* \)
    \[ \text{setscore}(G^*) = \sum_{G \subseteq G^*} \text{score}(G) \]
    \[ = \sum_{G \subseteq G^*} \prod_{i=1}^{n} l_s(i, G_i) \]
    \[ = \sum_{G_1 \subseteq G_1} \sum_{G_2 \subseteq G_2} \ldots \sum_{G_n \subseteq G_n} \prod_{i=1}^{n} l_s(i, G_i) \]
    \[ = \prod_{i=1}^{n} \sum_{G_i \subseteq G_i} l_s(i, G_i) \]
  - The fast zeta transform is a way to compute a set function \( F(V) = \sum_{V \subseteq U} f(V) \) for all sets \( V \subseteq U \) in \( O(n2^n) \) time.
when possible it works in practice, comparing
\[
P(d|D, α, 2^{G^*}) \quad \text{and} \quad P(d|D, α, G^*)
\]

<table>
<thead>
<tr>
<th>Data</th>
<th>BDeu 1.0</th>
<th>fNML</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostOperative</td>
<td>1.033 ± 0.052912</td>
<td>1.181 ± 0.046533</td>
</tr>
<tr>
<td>Iris</td>
<td>1.000 ± 0.000855</td>
<td>0.997 ± 0.004127</td>
</tr>
<tr>
<td>Wine</td>
<td>1.019 ± 0.014496</td>
<td>1.079 ± 0.022389</td>
</tr>
<tr>
<td>Glass</td>
<td>1.059 ± 0.030724</td>
<td>1.010 ± 0.009331</td>
</tr>
<tr>
<td>Thyroid</td>
<td>1.013 ± 0.008427</td>
<td>1.006 ± 0.003870</td>
</tr>
<tr>
<td>HeartStatlog</td>
<td>1.025 ± 0.013616</td>
<td>1.044 ± 0.016531</td>
</tr>
<tr>
<td>BreastCancer</td>
<td>0.997 ± 0.007784</td>
<td>1.039 ± 0.012359</td>
</tr>
<tr>
<td>HeartHungarian</td>
<td>1.008 ± 0.008601</td>
<td>1.047 ± 0.011981</td>
</tr>
<tr>
<td>HeartCleveland</td>
<td>1.006 ± 0.009991</td>
<td>1.042 ± 0.015379</td>
</tr>
<tr>
<td>Ecoli</td>
<td>0.998 ± 0.005293</td>
<td>1.008 ± 0.003127</td>
</tr>
<tr>
<td>Liver</td>
<td>1.004 ± 0.006049</td>
<td>1.017 ± 0.005205</td>
</tr>
<tr>
<td>Balance</td>
<td>1.000 ± 0.000000</td>
<td>1.000 ± 0.000000</td>
</tr>
<tr>
<td>BcWisconsin</td>
<td>1.008 ± 0.004476</td>
<td>1.020 ± 0.005698</td>
</tr>
<tr>
<td>Diabetes</td>
<td>1.002 ± 0.001896</td>
<td>1.006 ± 0.002771</td>
</tr>
</tbody>
</table>

Thus learn a bit too complex model and average over substructures!

Parameter free alternatives based on code length

\[
P_{NML}(D|S) = \frac{P(D|\hat{\theta}(D), S)}{\sum_{D'} P(D|\hat{\theta}(D), S)},
\]

where \(D'\) goes over all the data sets that are of the same size as \(D\).

- Impossible to naively compute for multivariate discrete data.
- But possible to compute for univariate categorical data.
- See, no hyperparameters.
- \(P_{NML}\) has the \(\text{minmax}_D\) regret property - never much worse than post hoc \(P(D|\hat{\theta}(D), S)\).
- \(fNML\) and \(qNML\) are attempts to use these ideas to derive practical approximations to \(P_{NML}\).
Factorized codings

\[
P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_3)
\]

- \(X_1\) and \(X_2\) being just blocks of binary and ternary data, this data can be coded (separately) by a Bayesian mixture or, better yet, by a normalized maximum likelihood code.

\[
P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_3)
\]

- \(X_3\) is then coded using 2x3=6 parts using 6 different codes, one per each possible value of \((X_1,X_2)\)-pair.
Factorized codings

\[ P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_3) \]

- \( X_3 \) is then coded using \( 2 \times 3 = 6 \) parts using 6 different codes, one per each possible value of \( (X_1,X_2) \)-pair.

\[ P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_3) \]

- \( X_3 \) is then coded using \( 2 \times 3 = 6 \) parts using 6 different codes, one per each possible value of \( (X_1,X_2) \)-pair.
Factorized codings

\[
P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_3)
\]

- \( X_4 \) is then coded in two parts, one per each possible value of \( X_3 \in \{1,2\} \).

**Example models**

- Breast cancer data with \( N = 70, n = 10 \)

59446 parameters

1928 parameters
quotient normalized maximum likelihood (qNML)

- Computing NML for the whole data D is not feasible because of the normalizer.
  - for our 24x4 example matrix we have to sum over $2^{24}3^{24}2^{24}4^{24}$ different data matrices.
- in qNML, like in fNML, we use the fact that there are ways to compute NML for a single categorical data vector efficiently.
  $$S_{qNML}(D; H) := \prod_{i=1}^{n} \frac{P_{NML}^{i}(D_{i}; D_{\neq i}; H)}{P_{NML}^{i}(D_{\neq i}; H)}$$
- to code several data columns with $P_{NML}^{i}$, we first collapse their value combinations into a single categorical value.

qNML - column $X_1$

- With no parents, the columns $X_1$ and $X_2$ are coded (separately) by a normalized maximum likelihood code.

qNML - column $X_2$

- Collapsing $(X_1, X_2, X_3)$, and $(X_1, X_2)$ to code them by $P_{NML}^{i}$.

qNML - column $X_3$

- With no parents, the columns $X_1$ and $X_2$ are coded (separately) by a normalized maximum likelihood code.
qNML - column $X_4$

- Collapsing $X_3$ and $X_4$ to code them by $P_{NML}^1$.

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$\begin{bmatrix}X_1 & X_2 & X_3 & X_4\end{bmatrix} = \begin{bmatrix}X_1 & X_2 & X_3 & X_4\end{bmatrix} \begin{bmatrix}a & b & c & d\end{bmatrix}$

Example models

- BDeu
- fNML
- qNML

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDeu</td>
<td>59446</td>
</tr>
<tr>
<td>fNML</td>
<td>1928</td>
</tr>
<tr>
<td>qNML</td>
<td>128</td>
</tr>
</tbody>
</table>

- Breast cancer data with $N = 70$, $n = 10$

Number of parameters by sample size

Predictive log-loss by sample size
Which parameters for fNML and qNML (and BIC/MDL)?

- It would be odd to suggest hyperparameters for parameter estimation if we argue for getting rid of them in structure learning
- Sequential normalized maximum likelihood
  - one step lookahead NML
    
    \[
    P(d | D, S) = \frac{P(d, D | \hat{\theta}(d, D), S)}{\sum_{d'} P(d', D | \hat{\theta}(d', D), S)}
    \]
  - For univariate categorical variable this leads to using the parameters:
    \[
    \theta_{ijk} \propto e(n_{jk})(n_{jk}+1),
    \]
    where \(e(n) = \left(\frac{n+1}{n}\right)^a\).
- Kind of weighted Laplace smoothing

**qNML pedictive results (bit bigger data)**

**Table:** Average predictive performance rank over different sample sizes for different model selection criteria in 20 different data sets.

<table>
<thead>
<tr>
<th>Data</th>
<th>N</th>
<th>BD(\text{Deu}^{\text{BPP}})</th>
<th>BIC(\text{sNML})</th>
<th>fNML(\text{sNML})</th>
<th>qNML(\text{sNML})</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostOpe</td>
<td>90</td>
<td>2.79</td>
<td>1.20</td>
<td>3.06</td>
<td>2.94</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>2.82</td>
<td>2.37</td>
<td>2.27</td>
<td>2.54</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>3.23</td>
<td>1.88</td>
<td>2.67</td>
<td>2.22</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>3.61</td>
<td>3.09</td>
<td>1.42</td>
<td>1.88</td>
</tr>
<tr>
<td>Thyroid</td>
<td>215</td>
<td>2.55</td>
<td>3.21</td>
<td>1.80</td>
<td>2.44</td>
</tr>
<tr>
<td>HeartSt</td>
<td>270</td>
<td>3.12</td>
<td>1.39</td>
<td>3.12</td>
<td>2.37</td>
</tr>
<tr>
<td>BreastC</td>
<td>286</td>
<td>3.09</td>
<td>1.41</td>
<td>2.97</td>
<td>2.53</td>
</tr>
<tr>
<td>HeartHu</td>
<td>294</td>
<td>3.18</td>
<td>1.66</td>
<td>2.90</td>
<td>2.27</td>
</tr>
<tr>
<td>HeartCl</td>
<td>303</td>
<td>3.46</td>
<td>1.38</td>
<td>2.99</td>
<td>2.17</td>
</tr>
<tr>
<td>Ecoli</td>
<td>336</td>
<td>3.20</td>
<td>3.53</td>
<td>1.24</td>
<td>2.04</td>
</tr>
<tr>
<td>Liver</td>
<td>345</td>
<td>3.17</td>
<td>2.39</td>
<td>2.69</td>
<td>1.75</td>
</tr>
</tbody>
</table>

**qNML pedictive results (small data)**

**Table:** Average predictive performance rank over different sample sizes for different model selection criteria in 20 different data sets.

<table>
<thead>
<tr>
<th>Data</th>
<th>N</th>
<th>BD(\text{Deu}^{\text{BPP}})</th>
<th>BIC(\text{sNML})</th>
<th>fNML(\text{sNML})</th>
<th>qNML(\text{sNML})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance</td>
<td>625</td>
<td>3.35</td>
<td>1.91</td>
<td>1.59</td>
<td>3.16</td>
</tr>
<tr>
<td>BcWisco</td>
<td>699</td>
<td>3.06</td>
<td>2.03</td>
<td>2.89</td>
<td>2.02</td>
</tr>
<tr>
<td>Diabete</td>
<td>768</td>
<td>2.91</td>
<td>2.70</td>
<td>2.68</td>
<td>1.71</td>
</tr>
<tr>
<td>TicTacT</td>
<td>958</td>
<td>3.44</td>
<td>2.71</td>
<td>1.31</td>
<td>2.53</td>
</tr>
<tr>
<td>Yeast</td>
<td>1484</td>
<td>2.60</td>
<td>3.76</td>
<td>1.55</td>
<td>2.10</td>
</tr>
<tr>
<td>Abalone</td>
<td>4177</td>
<td>2.60</td>
<td>3.64</td>
<td>1.04</td>
<td>2.72</td>
</tr>
<tr>
<td>PageBlo</td>
<td>5473</td>
<td>2.24</td>
<td>3.61</td>
<td>1.31</td>
<td>2.83</td>
</tr>
<tr>
<td>Adult</td>
<td>32561</td>
<td>3.23</td>
<td>3.77</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Shuttle</td>
<td>58000</td>
<td>1.44</td>
<td>3.78</td>
<td>1.56</td>
<td>3.22</td>
</tr>
</tbody>
</table>

**qNML pedictive results**

- qNML does not have tunable hyper parameters
- qNML yields more parsimonious models than BD\(\text{Deu}\) and fNML
- qNML is still competitive predictively
- qNML is as quick/slow to compute than BD\(\text{Deu}\) and fNML
- qNML gives equal scores to equal dependency hypotheses
- qNML is consistent and regular
- qNML coincides with NML for exponentially many models
Outline

- Preliminaries
- Motivation
- Information theoretic criteria
- Conclusion

Summary

- Out of “common” scoring criteria BIC and qNML have all the “desirable properties”
  - BIC is based on asymptotic approximation of marginal likelihood
    - or two part code (sometimes missing valued MDL)

<table>
<thead>
<tr>
<th></th>
<th>BIC</th>
<th>BDeu</th>
<th>BDs</th>
<th>fNML</th>
<th>qNML</th>
<th>NML</th>
</tr>
</thead>
<tbody>
<tr>
<td>decomposable</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>consistent</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>LH equivalent</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Regular</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Hyperparameter free</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Table: Features of scoring criteria

Conclusion

- The parameter sensitivity is not just about the selected network
  - but the whole posterior distribution
- The parameter sensitivity is still a problem
  - blessing
- I still do not know what model to give to a practitioner
  - but I tend to check the parameter free options first
  - and also check for the probability of other models to have an idea how probable my model is

Thank You!