bayesstats ic — Bayesian information criteria and Bayes factors

Description
bayesstats ic calculates and reports model-selection statistics, including the deviance information criterion (DIC), log marginal-likelihood, and Bayes factors (BFs), using current estimation results from the bayesmh command. BFs can be displayed in the original metric or in the log metric. The command also provides two different methods to approximate marginal likelihood.

Quick start
Information criteria for previously saved estimation results A and B with A used as the base model by default
   bayesstats ic A B
As above, but use B as the base model instead of A
   bayesstats ic A B, basemodel(B)
Report BFs instead of the default log BFs
   bayesstats ic A B, bayesfactor

Menu
Statistics > Bayesian analysis > Information criteria
Syntax

bayesstats ic [namelist] [, options]

namelist is a name, a list of names, _all, or *. A name may be ., meaning the current (active) estimates. _all and * mean the same thing.

options          Description

Main

basemodel(name)  specify a base or reference model; default is the first-listed model
bayesfactor     report BFs instead of the default log BFs
diconly         report only DIC

Advanced

marglmethod(method) specify marginal-likelihood approximation method; default is to use Laplace–Metropolis approximation, lmetropolis; rarely used

method          Description

lmetropolis     Laplace–Metropolis approximation; the default
hmean           harmonic-mean approximation

Options

Main

basemodel(name) specifies the name of the model to be used as a base or reference model when computing BFs. By default, the first-listed model is used as a base model.
bayesfactor specifies that BFs be reported instead of the default log BFs.
diconly specifies that only DIC be reported in the table and that the log marginal likelihood and Bayes factors be omitted from the table. Options basemodel(), bayesfactor, and marglmethod() have no effect when the diconly option is specified.

Advanced

marglmethod(method) specifies a method for approximating the marginal likelihood. method is either lmetropolis, the default, for Laplace–Metropolis approximation or hmean for harmonic-mean approximation. This option is rarely used.

Remarks and examples

Remarks are presented under the following headings:

Bayesian information criteria
Bayes factors
Using bayesstats ic
Bayesian information criteria

Bayesian information criteria are used for selecting a model among a set of candidate models that best fits the data. Likelihood-based inference is known to be prone to overfitting the data. Indeed, it is often possible to increase the likelihood by simply including more parameters in a model. Bayesian information criteria address this problem by applying a penalty proportional to the complexity of the models to the likelihood.

Consider a finite set of Bayesian models $M_1, \ldots, M_r$, which we want to compare with a base model $M_b$. All models $M_j$s are fit to the same dataset but may differ in their likelihood or prior specification.

Three commonly used information criteria are Akaike information criterion (AIC), Bayesian information criterion (BIC), and DIC. All three criteria are likelihood based and include a goodness-of-fit term proportional to the negative likelihood of the model and a penalty term proportional to the number of parameters in the model. Models with smaller values of these criteria are preferable.

The BIC, originally derived for the exponential family of distributions, is based on the assumption that the model has a flat, noninformative prior. In frequentist statistics, BIC is widely used as a variable-selection criterion, particularly in linear regression. In BIC, the penalty term is a product of the number of parameters in the model and the log of the sample size. The penalty of BIC thus increases not only with the number of parameters but also with the sample size. In the AIC, the penalty term is two times the number of parameters and does not depend on the sample size. As a result, BIC is more conservative than AIC and prefers simpler models.

DIC is similar to AIC, but its penalty term is based on a complexity term that measures the difference between the expected log likelihood and the log likelihood at the posterior mean point. DIC is designed specifically for Bayesian estimation that involves MCMC simulations.

The limitation of all three criteria is that they either ignore prior distributions or assume that prior distributions are noninformative. They are thus not well suited for Bayesian sensitivity analysis, when models with the same parameters but different priors are being compared.

The bayesstats ic command reports DIC. See [R] estat ic after the corresponding maximum likelihood estimation command for values of AIC and BIC.

Bayes factors

In Bayesian inference, BFs are preferred to model-selection criteria because, unlike BIC, AIC, and DIC, they incorporate the information about model priors. Taking into account prior information is essential for Bayesian sensitivity analysis, when models with the same parameters but different priors are being compared.

The BF of two models is just the ratio of their marginal likelihoods calculated using the same dataset. Unlike BIC, AIC, and DIC, BFs include all information about the specified Bayesian model. Thus BFs are not applicable to models with improper priors, whereas BIC, AIC, and DIC are still applicable because they ignore prior information. BFs, however, are often difficult to compute reliably because of the difficulty in computing marginal likelihoods.

BFs also require that posterior distributions be completely specified, including the normalizing constants. The latter is especially important in Bayesian estimation using MCMC simulations, when the normalizing constants are often omitted from the specification of a posterior distribution. The bayesmh command always simulates from a complete posterior distribution when you select one of the supported Bayesian models, but you need to make sure to include all normalizing constants with your posterior distribution when you are programming your own Bayesian model (see [BAYES] bayesmh evaluators) and would like to use BFs during postestimation.
Let $BF_{jb}, j = 1, \ldots, r$, be the BF of model $M_j$ with respect to the base model $M_b$. All models $M_j$ are fit to the same dataset; otherwise, BFs are meaningless. The `bayesstats ic` command calculates $BF_{jb}$'s and reports them in absolute metric or in log metric when the `bayesfactor` option is specified.

Jeffreys (1961) proposes the following interpretation of the values of $BF_{jb}$ based on half-units of the log metric:

<table>
<thead>
<tr>
<th>$\log_{10}(BF_{jb})$</th>
<th>$BF_{jb}$</th>
<th>Evidence against $M_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 1/2</td>
<td>1 to 3.2</td>
<td>Bare mention</td>
</tr>
<tr>
<td>1/2 to 1</td>
<td>3.2 to 10</td>
<td>Substantial</td>
</tr>
<tr>
<td>1 to 2</td>
<td>10 to 100</td>
<td>Strong</td>
</tr>
<tr>
<td>$&gt;2$</td>
<td>$&gt;100$</td>
<td>Decisive</td>
</tr>
</tbody>
</table>

Kass and Raftery (1995) suggest using twice the natural logarithm of the BF to make it have the same scale as the DIC and likelihood-ratio test statistic. They suggest the following interpretation table:

<table>
<thead>
<tr>
<th>$2 \log_e(BF_{jb})$</th>
<th>$BF_{jb}$</th>
<th>Evidence against $M_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 2</td>
<td>1 to 3</td>
<td>Bare mention</td>
</tr>
<tr>
<td>2 to 6</td>
<td>3 to 20</td>
<td>Positive</td>
</tr>
<tr>
<td>6 to 10</td>
<td>20 to 150</td>
<td>Strong</td>
</tr>
<tr>
<td>$&gt;10$</td>
<td>$&gt;150$</td>
<td>Very strong</td>
</tr>
</tbody>
</table>

Typically, the worst-fitting model is chosen as a base model. If the base model happens to be better than the comparison model, the corresponding BF will be negative. In this case, you can apply results above to the absolute value of the BF.

BFs compute relative probabilities of how well each model fits the data compared with the base model. Being relative quantities, BFs cannot be used to measure goodness of fit of a particular model unless one assumes that the base model fits the data well. Some researchers view this as a limitation of BFs (Gelman et al. 2014). Kass and Raftery (1995), on the other hand, show that BFs can be viewed as differences between predictive scores and thus can be used to measure success of different models at predicting the data.

BFs have several advantages over the more traditional, frequentist testing methods. For example, they do not have the limitation of the $p$-value approach to systematically reject the null hypothesis in large samples. BFs are also suitable for comparing both nonnested and nested models. Also see `Comparing Bayesian models` in [BAYES] intro for more information about Bayesian model comparison.

A key element in computing BFs is calculating the marginal likelihood. Except for some rare cases, marginal likelihood does not have a closed form and needs to be approximated. A detailed review of different approximation methods is given by Kass and Raftery (1995). The default method implemented in `bayesstats ic` (and `bayesmh`) is the Laplace–Metropolis approximation (Lewis and Raftery 1997). The harmonic-mean approximation of the marginal likelihood is also available via the `marglimethod(hmean)` option, but we recommend that you use the default method. See `Methods and formulas` in [BAYES] bayesmh for technical details.
Using bayesstats ic

Example 1

The `bayesstats ic` command provides several model-selection statistics that can be used to compare models. To illustrate the use of `bayesstats ic`, we consider `auto.dta`. We model the fuel-efficiency variable `mpg` using a normal distribution with fixed variance but unknown, random mean. There is only one random parameter in this model—`{mpg:_cons}`. We compare the models with three different prior distributions to find the best one among them. We fit the three models using `bayesmh` and save the corresponding estimation results as `uniform1`, `uniform2`, and `normal`.

First, for comparison purposes, let’s obtain the maximum likelihood estimate (MLE) of the mean of `mpg`, which is simply the sample mean in our example:

```stata
use http://www.stata-press.com/data/r14/auto
(1978 Automobile Data)
summarize mpg
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>74</td>
<td>21.2973</td>
<td>5.785503</td>
<td>12</td>
<td>41</td>
</tr>
</tbody>
</table>

The sample mean of `mpg` is roughly 21.3.

Next, we use `bayesmh` to fit our first model of interest. We fix the variance of the normal distribution to 30, which is close to the estimated variance of `mpg` of $5.79^2 = 33.52$.

```stata
set seed 14
bayesmh mpg, likelihood(normal(30))
> prior({mpg:_cons}, uniform(-10, 10))
> initial({mpg:_cons} 2) saving(uniform1_simdata)
Burn-in ...
Simulation ...
Model summary

Likelihood:
`mpg ~ normal({mpg:_cons},30)`
Prior:
`{mpg:_cons} ~ uniform(-10,10)`

Bayesian normal regression
Random-walk Metropolis-Hastings sampling
MCMC iterations = 12,500
Burn-in = 2,500
MCMC sample size = 10,000
Number of obs = 74
Acceptance rate = .4102

Log marginal likelihood = -397.42978
Efficiency = .08018

<table>
<thead>
<tr>
<th>mpg</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
</table>

file uniform1_simdata.dta saved
estimates store uniform1
```

In the first model, we deliberately chose a prior for `{mpg:_cons}`, `uniform(-10,10)`, that does not include the value of the sample mean. We thus expect this model to fit poorly. Because of the restricted domain of the specified uniform prior, we also needed to specify an initial value for `{mpg:_cons}` for MCMC to start from a point of positive posterior probability.
We also specified the `saving()` option to save the MCMC simulation dataset so that we could use `estimates store` to store our estimation results for future use. See *Storing estimation results after bayesmh* in [BAYES] `bayesmh postestimation` for details.

```
. set seed 14
. bayesmh mpg, likelihood(normal(30))
   > prior({_mpg:_cons}, uniform(10, 30))
   > initial({_mpg:_cons} 20) saving(uniform2_simdata)
Burn-in ...
Simulation ...
Model summary

Likelihood:
   mpg ~ normal({_mpg:_cons},30)

Prior:
   {_mpg:_cons} ~ uniform(10,30)

Bayesian normal regression
Random-walk Metropolis-Hastings sampling

Model summary

MCMC iterations = 12,500
Burn-in = 2,500
MCMC sample size = 10,000
Number of obs = 74
Acceptance rate = .4272
Log marginal likelihood = -237.08583
Efficiency = .2414

<table>
<thead>
<tr>
<th>mpg</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
</table>

file uniform2_simdata.dta saved
. estimates store uniform2

In the second model, we used a uniform prior that included the value of the sample mean in its domain.
In the third model, we used a normal prior with a variance fixed at 30. Note that we did not need to specify an initial value for \{mpg:_cons\} in this model, because the domain of the normal distribution is the whole real line.

Both the uniform2 and normal models yield estimates close to the MLE of 21.3. According to their credible intervals, the domain of the posterior distribution of \{mpg:_cons\} is concentrated around MLE. For example, the 95% credible interval for the uniform2 model is [20.06, 22.60].

Now, let’s use bayesstats ic to compare the three models. We list all the models following the command name and use the normal model as a reference model.

```
. bayesstats ic uniform1 uniform2 normal, basemodel(normal)
```

Bayesian information criteria

<table>
<thead>
<tr>
<th></th>
<th>DIC</th>
<th>log(ML)</th>
<th>log(BF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniform1</td>
<td>785.89</td>
<td>-397.43</td>
<td>-153.26</td>
</tr>
<tr>
<td>uniform2</td>
<td>471.19</td>
<td>-237.08</td>
<td>7.08</td>
</tr>
<tr>
<td>normal</td>
<td>471.39</td>
<td>-244.16</td>
<td></td>
</tr>
</tbody>
</table>

Note: Marginal likelihood (ML) is computed using Laplace-Metropolis approximation.

The uniform1 model performs worse than the other two models according to the log marginal-likelihood, log(ML), and DIC—the DIC value is much larger, and the log(ML) value is much smaller for the uniform1 model. The other two models have only slightly different values for DIC and log(ML), according to which the uniform2 model is preferable.

Although the uniform2 and normal models have different prior distributions, they have almost identical posterior domain, that is, the range of values of \{mpg:_cons\} where the posterior is strictly positive. As such, they will have the same values for AIC and BIC, and we will not be able to discriminate between the two models based on these information criteria.
The most decisive factor between the uniform2 and normal models is the BF. The value of log BF, log(BF), is 7.08, which provides very strong evidence in favor of the uniform2 model.

We thus conclude that uniform2 is the best model among the three considered models. This may be explained by the fact that the specified uniform(10,30) prior is in more agreement with the likelihood of the data than the specified normal(0,30) prior.

After your analysis, remember to erase the saved simulation datasets you no longer need. For example, we erase all of them by typing

`. erase uniform1_simdata.dta
`. erase uniform2_simdata.dta
`. erase normal_simdata.dta

Stored results
bayesstats ic stores the following in r():

Scalars
r(bayesfactor) 1 if bayesfactor is specified; 0 otherwise

Macros
r(names) names of estimation results used
r(basemodel) name of the base or reference model
r(marglmethod) method for approximating marginal likelihood: lmetropolis or hmean

Matrices
r(ic) matrix reporting DIC, log(ML), and log(BF) or BF if bayesfactor is used

Methods and formulas
DIC was introduced by Spiegelhalter et al. (2002) for Bayesian model selection using MCMC simulations. DIC is based on the deviance statistics

\[ D(\theta) = -2 \{ \log f(y; \theta) - \log f^*(y; \theta^*) \} \]

where \( f(\cdot; \cdot) \) is the likelihood function of the model and \( f^*(y; \theta^*) \) is the likelihood of the full model that fits data perfectly. Because \( f^*(y; \theta^*) \) is constant across models fit to the same data, it is ignored in the actual calculation of DIC. Given an MCMC sample \( \{ \theta_t \}_{t=1}^T \), the expected deviance can be estimated by the sample average \( \overline{D}(\theta) = 1/T \sum_{t=1}^T D(\theta_t) \). Similarly to AIC and BIC, DIC is a sum of two components: the goodness-of-fit term \( \overline{D}(\theta) \) and the model complexity term \( p_D \): DIC = \( \overline{D}(\theta) + p_D \). The complexity is defined as the difference between the expected deviance and the deviance at the sample posterior mean: \( p_D = \overline{D}(\theta) - D(\theta) \). We thus have

\[ \text{DIC} = D(\overline{\theta}) + 2p_D \]

Models with smaller values of DIC are preferred to models with larger values of DIC.

BFs were introduced by Jeffreys (1961). The BF of two models, \( M_1 \) and \( M_2 \), is given by

\[ \text{BF}_{12} = \frac{P(y|M_1)}{P(y|M_2)} = \frac{m_1(y)}{m_2(y)} \]
where \( m_1(\cdot) \) and \( m_2(\cdot) \) are the corresponding marginal likelihoods associated with models \( M_1 \) and \( M_2 \). (See Methods and formulas in [BAYES] bayesmh for details about computing marginal likelihood.) BF\(s \) are defined only for proper marginal densities. Comparing models with improper priors is allowed as long as the resulting marginal densities are proper. The methodological importance of BF\(s \) comes from the fact that the so-called posterior odds is a product of prior odds and BF:

\[
\frac{P(M_1|y)}{P(M_2|y)} = \frac{P(M_1)}{P(M_2)} \times BF_{12}
\]

Therefore, if we assume that \( M_1 \) and \( M_2 \) are equally probable a priori, the posterior odds will be equal to the BF. We thus prefer model \( M_1 \) if \( BF_{12} > 1 \) and model \( M_2 \) otherwise. In practice, because of the higher numerical stability, we often calculate BF\(s \) in the (natural) log metric and compare its value against 0.

\[
\log BF_{12} = \log m_1(y) - \log m_2(y)
\]

References


Also see

[BAYES] bayesmh — Bayesian regression using Metropolis–Hastings algorithm

[BAYES] bayesmh postestimation — Postestimation tools for bayesmh

[BAYES] bayestest model — Hypothesis testing using model posterior probabilities

[R] estimates — Save and manipulate estimation results