Title

bmastats lps — Log predictive-score after BMA regression

Description

bmastats lps computes and optionally stores in a new variable the log predictive-score (LPS) for observations after the bmaregress command. LPS is used to assess the predictive performance of Bayesian model averaging (BMA) models.

Quick start

Compute and summarize LPS for all observations

    bmastats lps

Same as above, but use a known variance to compute and report the entropy

    bmastats lps, sigma2(0.1)

Compute LPS for the first 12 observations, and store them in newvar

    bmastats lps in 1/12, generate(newvar)

Compute LPS for previously stored bma1 and bma2 estimation results, and store respective LPS in variables newvar1 and newvar2

    bmastats lps bma1 bma2, generate(newvar*)

Display multiple estimation results compactly in one table

    bmastats lps bma1 bma2, compact

After fitting bmaregress using a training sample, sample == 1, compute LPS for out-of-sample observations—test sample sample == 2—to compare out-of-sample predictive performance of two BMA models with estimation results stored in bma1 and bma2

    bmastats lps bma1 bma2 if sample == 2

Menu

Statistics > Bayesian model averaging > Log predictive-score

Syntax

Summarize LPS for current estimation results

> bmastats lps [*if*] [*in*] [, <u>sigma2</u>(*#*) <u>gen</u>erate(*newvar*[, double])]

Summarize LPS for multiple estimation results

> bmastats lps *namelist* [*if*] [*in*] [, <u>sigma2</u>(*#* [*#* [...]])
>
> <u>gen</u>erate(*newvarspec*[, double]) compact]

namelist includes names of previously stored estimation results after bmaregress. The current (active) results are the default.

newvarspec is a *newvar* for a single estimation result or *newvarlist* or *stub** for multiple estimation results.

collect is allowed; see [U] **11.1.10 Prefix commands**.

Options

> ┌─────┐
> │ Main │
> └─────┘

<u>sigma2</u>(*#* [*#* [...]]) specifies the variance to be used to compute the entropy. With out-of-sample observations, the posterior mean estimate of the error variance from bmaregress is used. With multiple estimation results, you can specify just one variance value to be used for all estimation results or specific values for each result. A missing value (.) in place of a *#* means that the default variance estimate is to be used for the corresponding estimation result. This option is useful to compute the entropy more precisely when the variance is known, such as during simulation studies.

<u>gen</u>erate(*newvarspec*[, double]) specifies the name (or names) of new variables to store the LPS values. A new variable name is specified with one estimation result. A list of new variable names or *stub** is specified with multiple estimation results.

> double specifies that the new variables be stored as double. If the double suboption is not specified, the variables are stored using the current type as set by set type (see [D] **generate**), which is float by default.

compact specifies that multiple estimation results be displayed in a compact form. By default, a separate header and table are displayed for each estimation result. If option compact is specified, the results are displayed in one table with a common header. If the variances differ across multiple estimation results, the variance and entropy are not reported in the header. This option requires at least two estimation results.

Remarks and examples

stata.com

To evaluate the predictive performance of a model, one needs a measure that summarizes the quality of predictions (Piironen and Vehtari 2017). Good (1952) suggests that such a measure be defined based on the posterior predictive distribution. The LPS is defined as the negative of the log of the posterior predictive density evaluated at an observation. The LPS for out-of-sample observations can be used to evaluate the predictive performance of models. Fernández, Ley, and Steel (2001b) were among the first to use the LPS to evaluate the predictive performance of BMA. And Ley and Steel (2012) compared a variety of *g*-priors for the linear BMA using the LPS.

As pointed out in Fernández, Ley, and Steel (2001a), the LPS is related to the Kullback–Leibler divergence between the actual sampling density and the out-of-sample predictive density. In theory, the expected LPS is bounded from below by the entropy of the sampling density, but in practice, this may not always hold because the entropy itself is being estimated. The closer the expected value is to the entropy, the more accurate the prediction. For the normal linear model, the entropy is given by $0.5\{\log(2\pi\sigma^2) + 1\}$, where σ^2 is the error variance.

bmastats lps reports the mean, minimum, and maximum of LPS for each specified estimation result. By default, the command reports the LPS summaries computed using the current dataset that includes in-sample observations, which were used during estimation and are identified by e(sample) == 1, for the current or active estimation results. These summaries are useful when comparing multiple models. By themselves, their values are not directly interpretable, unless the entropy is known or can be estimated reliably. In this case, the closer the mean LPS is to the entropy, the better the model fit. For instance, if you are using bmastats lps with a simulated dataset, you can specify the known error variance in the sigma2() option, and the corresponding entropy value will be reported. You can compare this value with the mean LPS to check model fit. When bmastats lps is used with only out-of-sample observations, observations not used to fit the model for which e(sample) == 0, the command uses the posterior mean estimate of the error variance from the fitted bmaregress model to estimate and report the entropy. You can then compare the out-of-sample mean LPS with the estimated entropy to check the out-of-sample model fit, assuming that the entropy estimate is a reliable estimate of the entropy of the true model. The entropy estimate is not reported when bmastats lps is used with in-sample observations.

You can use the generate() option to store observation-specific LPS values in a new variable. This is equivalent to using bmapredict, lps; see [BMA] **bmapredict**. With multiple estimation results, you must specify the same number of new variable names or one variable stub in generate().

With multiple estimation results, bmastats lps displays a separate table for each result. You can use the compact option to display the results in one table, which is more convenient for comparison.

Below, we demonstrate a more common usage of bmastats lps for comparing out-of-sample predictive performance of multiple BMA models.

▷ Example 1: Prediction of systolic blood pressure

Consider the chd dataset from Hastie, Tibshirani, and Friedman (2009), which is a subset of the coronary heart disease dataset used in Rossouw et al. (1983). It contains information about potential heart disease factors such as age, adiposity, sbp (systolic blood pressure), tobacco (consumption of), ldl (low density liboprotein), and others.

```
. use https://www.stata-press.com/data/r18/chd
(Coronary risk factor data from rural communities)

. describe

Contains data from https://www.stata-press.com/data/r18/chd.dta
 Observations:             462                  Coronary risk factor data from
                                                  rural communities
     Variables:             10                  7 Mar 2023 09:19
                                                (_dta has notes)
────────────────────────────────────────────────────────────────────────────
Variable       Storage   Display    Value
    name          type    format    label      Variable label
────────────────────────────────────────────────────────────────────────────
sbp             int      %8.0g                 Systolic blood pressure
tobacco         float    %6.2f                 Yearly tobacco use (kg)
ldl             float    %6.2f                 Low-density lipoprotein (LDL)
adiposity       float    %6.2f                 Body adiposity index
famhist         byte     %9.0g      noyes      Family history of ischaemic heart
                                                 disease
typea           byte     %8.0g                 Type A personality score
obesity         float    %6.2f                 Obesity (body mass index)
alcohol         float    %7.2f                 Alcohol use (grams/day)
age             byte     %8.0g                 Age (years)
chd             byte     %8.0g      noyes      Diagnosed with CHD
────────────────────────────────────────────────────────────────────────────
Sorted by:
```

We wish to use BMA to build a prediction model for systolic blood pressure using all other available factors, `tobacco` through `age`, as predictors. We consider the log-transformed `sbp` variable, `lsbp`, to make its distribution closer to normal. We also split the sample into five groups to have training and test subsamples to evaluate the predictive performance of a model. Because the sample is split randomly, we specify a random-number seed with `splitsample` for reproducibility.

```
. splitsample, generate(sample) nsplit(5) rseed(100)

. generate double lsbp = log(sbp)
```

We will use the first four subsamples as our training data to fit a model. And we will check the model's performance using the fifth, test, subsample. To demonstrate, we fit the `bmaregress` command using the default settings.

```
. bmaregress lsbp tobacco-age if sample != 5

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                No. of obs          =      370
Linear regression                       No. of predictors   =        8
Model enumeration                                  Groups   =        8
                                                   Always   =        0
Priors:                                 No. of models       =      256
  Models: Beta-binomial(1, 1)               For CPMP >= .9   =        7
   Cons.: Noninformative                Mean model size      =    2.132
   Coef.: Zellner's g
       g: Benchmark, g = 370            Shrinkage, g/(1+g)  = 0.9973
  sigma2: Noninformative                Mean sigma2          =    0.015
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0020223	.0008055	8	.93569
adiposity	.0042121	.0016916	3	.92348
obesity	.0004441	.001573	6	.10111
typea	−.0000344	.0002133	5	.047293
alcohol	9.80e−06	.0000733	7	.03971
tobacco	−.0000101	.0002961	1	.028973
ldl	−9.92e−06	.0005911	2	.02786
famhist	.0000648	.0022977	4	.027833
Always				
_cons	4.71339	.0366564	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 256 models.
Note: Default priors are used for models and parameter g.
```

With a small number of predictors, the model space is explored fully by model enumeration. The default fixed `benchmark` g-prior is used, which results in a negligible shrinkage factor of almost 1. The posterior mean size of the models is 2.1, and all potential predictors have PIPs greater than 2%.

We use the `bmastats lps` command to evaluate the prediction performance of the fit model on the test sample by using the condition `if sample == 5`.

```
. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =   .015329
Entropy                = −.670055
```

	Mean	Minimum	Maximum
LPS	−.5183101	−1.169732	4.861686

```
Note: Using analytical PMPs.
```

The command reports the number of predicted observations, the estimated entropy, and the variance estimate used to compute the entropy in the header. The output table reports the mean, minimum, and maximum LPS values.

The mean LPS of -0.52 is, as expected, greater than the entropy estimate, -0.67. The closer it is to the entropy, the better the predictive performance. The actual LPS values cannot be interpreted, but we can use them to compare the predictive performances among several BMA models.

Based on their PIPs above 90%, `age` and `adiposity` appear to be the more important predictors

of the elevated blood pressure (at least based on our training sample). We may ask what would be the performance of the model using only these two predictors?

To fit just one model of interest using bmaregress, we can specify all predictors of interest as one always-included group as follows. (In this case, the results from bmaregress will be similar to those from regress; see [R] **regress**.)

```
. bmaregress lsbp (age adiposity, always) if sample != 5
Enumerating models ...
Computing model probabilities ...
```

Bayesian model averaging				No. of obs	=	370
Linear regression				No. of predictors	=	2
Model enumeration				Groups	=	0
				Always	=	2
Priors:				No. of models	=	1
Models: Beta-binomial(1, 1)				For CPMP >= .9	=	1
Cons.: Noninformative				Mean model size	=	2.000
Coef.: Zellner's g						
g: Benchmark, g = 370				Shrinkage, g/(1+g)	=	0.9973
sigma2: Noninformative				Mean sigma2	=	0.015

lsbp	Mean	Std. dev.	Group	PIP
Always				
age	.002073	.0005608	0	1
adiposity	.0044281	.0010493	0	1
_cons	4.715601	.0236217	0	1

Note: Coefficient posterior means and std. dev. estimated from 1 model.
Note: Default priors are used for models and parameter g.

bmaregress visits only one model: regress lsbp age adiposity. In this case, the prior model probability has no effect on the BMA results, and bmaregress is equivalent to fitting a Bayesian linear model with Zellner's g-prior on regression coefficients with $g = 370$ and Jeffreys prior for the error variance parameter; see [BAYES] **bayes: regress** or [BAYES] **bayesmh**.

Let's calculate the mean LPS score for this model on the test sample.

```
. bmastats lps if sample == 5
Log predictive-score (LPS)
Number of observations =        92
Posterior mean sigma2  =   .015277
Entropy                =  -.67177
```

	Mean	Minimum	Maximum
LPS	-.5173158	-1.171843	4.890529

Note: Using analytical PMPs.

The mean LPS score of -0.52 is about the same as that of the full BMA regression, which suggests a similar predictive performance for the two models.

We can also investigate the impact of a particular factor on the prediction performance of our BMA model. For instance, in the first model, the tobacco usage term had a PIP of about 3%. We may investigate further whether the tobacco usage indeed has a relatively low impact on systolic blood pressure. To do this, we can fit BMA to all predictors except tobacco and compare its predictive performance to the full model.

```
. bmaregress lsbp ldl-age if sample != 5

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                      No. of obs        =       370
Linear regression                             No. of predictors =         7
Model enumeration                                      Groups   =         7
                                                       Always   =         0
Priors:                                       No. of models     =       128
  Models: Beta-binomial(1, 1)                     For CPMP >= .9 =         6
  Cons.: Noninformative                       Mean model size   =     2.150
  Coef.: Zellner's g
       g: Benchmark, g = 370                  Shrinkage, g/(1+g) =   0.9973
  sigma2: Noninformative                      Mean sigma2       =     0.015
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0020391	.0007811	7	.94501
adiposity	.0041866	.0016762	2	.92399
obesity	.0004608	.0015953	5	.10886
typea	−.0000412	.0002329	4	.056801
alcohol	.0000118	.0000802	6	.04785
ldl	−.0000121	.0006497	1	.033757
famhist	.0000758	.0025216	3	.03369

Always				
_cons	4.713174	.0370391	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 128 models.
Note: Default priors are used for models and parameter g.

. bmastats lps if sample == 5

Log predictive-score (LPS)

Number of observations =        92
Posterior mean sigma2  =   .015323
Entropy                =  −.670276
```

	Mean	Minimum	Maximum
LPS	−.5186576	−1.169858	4.855259

```
Note: Using analytical PMPs.
```

The mean LPS score of −0.52 is again close to that of the full model. This suggests that excluding the tobacco predictor does not diminish the predictive performance of the model. It does not appear to be a strong predictor in this dataset.

To more easily compare the LPS from the above models, we could have stored the estimation results from each model first and then specified these results with bmastats lps. To store the estimation results, we must use the saving() option with bmaregress. For instance,

```
. quietly bmaregress lsbp tobacco-age if sample != 5, saving(bma1)
. estimates store bma1
. quietly bmaregress lsbp (age adiposity, always) if sample != 5, saving(bma2)
. estimates store bma2
. quietly bmaregress lsbp ldl-age if sample != 5
```

```
. bmastats lps bma1 bma2 . if sample == 5, compact
```

Log predictive-score (LPS)

Number of observations = 92

LPS	Mean	Minimum	Maximum
bma1	−.5183101	−1.169732	4.861686
bma2	−.5173158	−1.171843	4.890529
Active	−.5186576	−1.169858	4.855259

Notes: Using analytical PMPs.
       Result **Active** has the smallest mean LPS.

We also specified the compact option to display the results in one table with a common header. Because the variance estimates differ between the estimation results, the variance and entropy are not shown in the header. (You can see them if you do not specify compact.) If the variances were the same, the variance and the entropy would have been displayed.

Remember to remove the generated datasets if they are no longer needed.

```
. erase bma1.dta
. erase bma2.dta
```

◁

In addition to the regression-function specification, in BMA, it is also important to evaluate the effect of different prior specifications on the results. Depending on how much information there is in the data about the data-generating process, different prior specifications may lead to different conclusions. We explore this in example 2.

▷ Example 2: Comparing different g-priors using cross-validation

For some data, the performance of linear BMA may depend strongly on the choice of the prior for regression coefficients (Fernández, Ley, and Steel 2001a). The classical Zellner's g-prior is commonly used in BMA for regression coefficients. The g parameter of this prior can be fixed or random. For a random g, a hyperprior for g must be specified. With a variety of prior choices available for g, the BMA specification can be challenging. As an objective criterion for choosing a prior, we can use a model's predictive performance on the data not used for fitting the model. In this example, we show how we can use cross-validation of log predictive-score to compare different choices of priors.

Let's continue with the analysis of the chd dataset from example 1. Here we will compare five different priors for g: two fixed and three random. The two fixed priors are the default benchmark, bench, and empirical Bayes local, ebl. The former uses a constant g-value for all models, whereas the latter uses a model-specific g. The three random priors are the benchmark beta-shrinkage, betabench(1); hyper-g/n, hypergn(3); and robust, robust; see *Priors for parameter g* in *Methods and formulas* of [BMA] **bmaregress** for details.

In example 1, we randomly split our sample into five subsamples. For cross-validation, we will treat each of five subsamples as a test sample and fit each of five models to the remaining four subsamples, which will give us 25 results. For each model i and test sample j, we will save the mean LPS in a 5-by-5 matrix.

To avoid manually fitting 25 models, we will automate this task by writing a simple program later. But first, let's go over one step of this automation manually. We will use a robust prior for g and, as in example 1, withhold the fifth sample for testing.

We use the gprior(robust) option to specify the robust g-prior and, because this is a random prior, use the rseed() option for reproducibility.

```
. bmaregress lsbp tobacco-age if sample != 5, gprior(robust) rseed(18)
```
```
Burn-in ...
Simulation ...
Computing model probabilities ...
```

Bayesian model averaging   No. of obs   =   370
Linear regression   No. of predictors =   8
MC3 and adaptive MH sampling   Groups =   8
                                             Always =   0
                                    No. of models   =   78
                                     For CPMP >= .9 =   9
Priors: Mean model size   =   2.366
  Models: Beta-binomial(1, 1)   Burn-in   =   2,500
   Cons.: Noninformative   MCMC sample size   = 10,000
   Coef.: Zellner's g   Acceptance rate   = 0.3419
      g: Robust
  sigma2: Noninformative   Mean sigma2   =   0.015

Sampling correlation = 0.9974

lsbp	Mean	Std. dev.		Group	PIP
age	.0020676	.0007043		8	.9688
adiposity	.0041715	.0015489		3	.9442
obesity	.0004562	.0015458		6	.1378
typea	−.0000531	.0002624		5	.0739
ldl	−.0000286	.0009378		2	.0716
alcohol	.0000163	.0000937		7	.0665
famhist	.0001123	.0032103		4	.0555
tobacco	−.000021	.0003703		1	.0474
Always					
_cons	4.713145	.0364241		0	1

Note: Coefficient posterior means and std. dev. estimated from 78 models.
Note: Default prior is used for models.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
g	325.2053	671.8658	18.0907	185.5779	82.95733	1396
Shrinkage	.9945226	.0029212	.000183	.9946403	.9880892	.9992842

The results are somewhat similar to those from the first model in example 1, except g is now random and Markov chain Monte Carlo (MCMC) sampling is used to estimate its posterior distribution.

```
. bmastats lps if sample == 5
```
```
Log predictive-score (LPS)
```
Number of observations =   92
Posterior mean sigma2   =   .015309
Entropy   = −.670731

	Mean	Minimum	Maximum
LPS	−.5191868	−1.170108	4.839892

Note: Using frequency PMPs.

```
. return list
scalars:
                  r(N) =  92
             r(sigma2) =  .0153086079526961
            r(entropy) =  -.6707314655467329
macros:
            r(pmptype) : "frequency"
matrices:
            r(summary) :  1 x 3
. matrix list r(summary)
r(summary)[1,3]
            Mean      Minimum      Maximum
LPS  -.51918683  -1.1701082     4.839892
. display el(r(summary),1,1)
-.51918683
```

After running bmastats lps, we use return list to display the stored results. The LPS summary
is stored in the r(summary) matrix. We can access its first element, the mean LPS, by using the
el() Stata matrix function.

Let's now automate the above in a program.

```
. program bma_cv
  1.          args matlps i gprior extraopts
  2.          forvalues j = 1/5 {
  3.                  bmaregress lsbp tobacco-age if sample != `j', ///
  >                                             gprior(`gprior') `extraopts'
  4.                  bmastats lps if sample == `j'
  5.                  matrix `matlps'[`i',`j'] = el(r(summary),1,1)
  6.          }
  7. end
```

The bma_cv program takes four arguments: a matrix name matlps to store the mean LPS values,
the matrix row index i, the prior specification gprior, and any additional options extraopts for
bmaregress, such as rseed() for the random g-priors. We then perform cross-validation by looping
over the five subsamples and withholding each one of them for testing. For each prior i, we save
the mean LPS for each cross-validation sample j in row `i' and column `j' of the specified matrix
`matlps'.

We create a 5-by-5 matrix lpscvmat to store the mean LPS values and call the bma_cv program
for each prior specification. Because the last three priors are random and thus use MCMC sampling
to estimate the results, the program may take a few moments longer to run compared with the one
for the fixed priors.

```
. matrix lpscvmat = J(5, 5, .)
. quietly bma_cv lpscvmat 1 "bench"
. quietly bma_cv lpscvmat 2 "ebl"
. quietly bma_cv lpscvmat 3 "betabench 1" "rseed(18)"
. quietly bma_cv lpscvmat 4 "hypergn 3" "rseed(18)"
. quietly bma_cv lpscvmat 5 "robust" "rseed(18)"
. matrix rownames lpscvmat = "bench" "ebl" "betabench 1" "hypergn 3" "robust"
. matrix colnames lpscvmat = "CV1" "CV2" "CV3" "CV4" "CV5"
```

```
. matlist lpscvmat
```

	CV1	CV2	CV3	CV4	CV5
bench	-.5771455	-.6311993	-.6583148	-.7233847	-.5183101
ebl	-.5778856	-.6339036	-.6623052	-.7182023	-.5224309
betabench 1	-.5774324	-.6320177	-.6591742	-.7232513	-.5199717
hypergn 3	-.5774459	-.6322509	-.6600753	-.7212745	-.5203261
robust	-.5758851	-.6344241	-.6594026	-.7210064	-.5191868

Smaller mean LPS scores indicate better predictive performance. All mean LPSs are rather similar, which suggests a similar predictive performance for the models with considered priors (assuming a similar variability between LPSs).

We can use a Mata matrix function, `colmin()`, to compute the smallest mean LPS value among all models for each cross-validated sample.

```
. mata: colmin(st_matrix("lpscvmat"))
                   1                 2                 3                 4

    1     -.5778855962     -.6344241355     -.6623052219     -.723384674

                   5

    1     -.5224308762
```

The second model, `ebl`, has the lowest mean LPS for the majority of models.

Let's fit a model using the `ebl` *g*-prior to the entire dataset.

```
. bmaregress lsbp tobacco-age, gprior(ebl)

Enumerating models ...
Computing model probabilities ...

Bayesian model averaging                      No. of obs         =      462
Linear regression                             No. of predictors  =        8
Model enumeration                                        Groups  =        8
                                                         Always   =        0
Priors:                                       No. of models      =      256
  Models: Beta-binomial(1, 1)                      For CPMP >= .9 =       46
   Cons.: Noninformative                     Mean model size     =    3.516
   Coef.: Zellner's g
       g: Empirical Bayes (local)            Shrinkage, g/(1+g) = 0.9660
  sigma2: Noninformative                     Mean sigma2         =    0.016
```

lsbp	Mean	Std. dev.	Group	PIP
age	.0026375	.0006026	8	.99981
adiposity	.0024261	.0016807	3	.7727
alcohol	.0003029	.0003148	7	.58378
obesity	.0014017	.0021937	6	.40683
tobacco	.0001807	.0007427	1	.2009
typea	-.0000758	.0003079	5	.19862
famhist	-.0005746	.0052701	4	.17773
ldl	-8.63e-06	.0013313	2	.17572
Always				
_cons	4.706904	.0433882	0	1

```
Note: Coefficient posterior means and std. dev. estimated from 256 models.
Note: Default prior is used for models.
```

Using the full dataset and the `ebl` prior, age is still a highly important predictor with a PIP of close to 1. The PIP of 0.77 for adiposity is now smaller, but the PIPs of 0.58 and 0.41 for alcohol and obesity, respectively, are now noticeably higher. The other predictors have slightly higher PIPs too, but they are still somewhat weak predictors relative to the other four.

If we run models using the other considered priors, we will find that our conclusions about the relative importance of predictors are almost the same. Of course, the PIP estimates themselves will be different because different priors impose different levels of shrinkage for the coefficients.

Beware that the results will also change if you use a different random-number seed. But for stable datasets and models, changing the random-number seed should not affect the conclusions.

◁

Stored results

bmastats lps stores the following in `r()` with one estimation result:

Scalars
  r(N)   number of observations
  r(sigma2)   variance used to compute entropy
  r(entropy)   entropy estimate

Macros
  r(name)   name of the estimation result (if any specified)
  r(pmptype)   analytical or frequency

Matrices
  r(summary)   summary matrix with mean, minimum, and maximum LPS

bmastats lps stores the following in `r()` for each #th estimation result:

Scalars
  r(N#)   number of observations
  r(sigma2_#)   variance used to compute entropy
  r(entropy#)   entropy estimate

Macros
  r(name#)   name of the #th estimation result
  r(pmptype#)   analytical or frequency

Matrices
  r(summary#)   summary matrix with mean, minimum, and maximum LPS

bmastats lps, compact stores the following in `r()` with multiple estimation results:

Macros
  r(names)   names of estimation results
  r(pmptypes)   type of PMP for each model: analytical or frequency

Matrices
  r(N)   numbers of observations
  r(sigma2)   variances used to compute entropy
  r(entropy)   entropy estimates
  r(summary)   summary matrix with mean, minimum, and maximum LPS for all estimation results

Methods and formulas

Let y^* be a new observation of the outcome and $\mathbf{x}^*$ be a p-vector of new values for the predictors, and let $\mathbf{z}^* = \mathbf{x}^* - \overline{\mathbf{x}}$, where $\overline{\mathbf{x}}' = \mathbf{1}'\mathbf{X}/n$ is a $p \times 1$ vector of predictor means. Let $\mathbf{y}$ be an n-vector of observed outcome values. LPS for y^* is defined as

$$\text{LPS}_{y^*} = -\log\{f(y^*|\mathbf{y}, \mathbf{z}^*)\}$$

For a BMA linear model with a Zellner's g-prior for the regression coefficients and fixed g, $f(y^*|\mathbf{y}, \mathbf{z}^*)$ is a 2^p mixture of t distributions. In this case, LPS can be calculated exactly, if we enumerate the entire model space (Fernández, Ley, and Steel 2001a),

$$\text{LPS}_{y^*} = -\log\left\{\sum_{j=1}^{2^p} P_a(M_j|\mathbf{y})f(y^*|M_j, \mathbf{y}, \mathbf{z}_j^*)\right\}$$

where $P_a(M_j|\mathbf{y})$'s are the analytical PMPs defined by (7) in *Posterior model probability* in *Methods and formulas* of [BMA] **bmaregress** and $f(y^*|M_j, \mathbf{y}, \mathbf{z}_j^*)$ is the conditional posterior predictive distribution, which is a t-distribution with $(n-1)$ degrees of freedom and location and scale given by (5) and (6) in *Conditional posterior predictive distribution* in *Methods and formulas* of [BMA] **bmaregress** with $q = 1$.

With fixed g and when model sampling is used, such as when bmaregress's `sampling` option is specified, the above formula is computed with respect to the set $J \subset \{1, 2, \ldots, 2^p\}$ of distinct models visited by the MCMC sampler.

With random g, we have an MCMC sample $\{g_t, m_t\}$ of size T from the joint g and model posterior distribution. Then, $f(y^*|\mathbf{y}, \mathbf{z}^*)$ in the definition of LPS is replaced with the following frequency-based MCMC estimator,

$$\widehat{f}(y^*|\mathbf{y}, \mathbf{z}^*) = \frac{1}{T}\sum_{t=1}^{T} f(y^*|g_t, m_t, \mathbf{y}, \mathbf{z}^*)$$

where $f(y^*|g_t, m_t, \mathbf{y}, \mathbf{z}^*)$ is the density of the location-scale t distribution with the respective mean and scale given by (5) and (6) in *Conditional posterior predictive distribution* in *Methods and formulas* of [BMA] **bmaregress** with $q = 1$ and j substituted for t. The bmastats lps command thus computes the following frequency-based LPS estimator:

$$\widehat{\text{LPS}}_{y^*} = -\log\{\widehat{f}(y^*|\mathbf{y}, \mathbf{z}^*)\}$$

Fernández, Ley, and Steel (2001b) and Ley and Steel (2012) suggest exploring the summary statistics of LPS such as the mean, minimum, and maximum to evaluate the quality of the predictions.

The entropy of a continuous distribution with density $f(x)$ is defined to be $-E[\log\{f(x)\}]$. For a normal distribution with mean μ and variance σ^2, the entropy is

$$\frac{1}{2}\{1 + \log(2\pi\sigma^2)\}$$

bmastats lps reports the estimated entropy with σ^2 in the above replaced by the estimated posterior mean variance as described by (13) and (14) in *Posterior inclusion probability* in *Methods and formulas* of [BMA] **bmaregress**.

References

Fernández, C., E. Ley, and M. F. J. Steel. 2001a. Benchmark priors for Bayesian model averaging. *Journal of Econometrics* 100: 381–427. https://doi.org/10.1016/S0304-4076(00)00076-2.

——. 2001b. Model uncertainty in cross-country growth regressions. *Journal of Applied Econometrics* 16: 563–576. https://doi.org/10.1002/jae.623.

Good, I. J. 1952. Rational decisions. *Journal of the Royal Statistical Society, Series B* 14: 107–114. https://doi.org/10.1111/j.2517-6161.1952.tb00104.x.

Hastie, T. J., R. J. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. New York: Springer.

Ley, E., and M. F. J. Steel. 2012. Mixtures of g-priors for Bayesian model averaging with economic applications. *Journal of Econometrics* 171: 251–266. https://doi.org/10.1016/j.jeconom.2012.06.009.

Piironen, J., and A. Vehtari. 2017. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing* 27: 711–735. https://doi.org/10.1007/s11222-016-9649-y.

Rossouw, J. J., J. E. du Plessis, J. P. Benadé, A. J. S. Jordaan, P. C. J. Kotzé, J. P. Jooste, and P. L. Ferreira. 1983. Coronary risk factor screening in three rural communities. The CORIS baseline study. *South African Medical Journal* 64: 430–436. https://doi.org/10.10520/AJA20785135_9894.

Also see

[BMA] **bmastats** — Summary for models and predictors after BMA regression

[BMA] **bmaregress** — Bayesian model averaging for linear regression

[BMA] **BMA postestimation** — Postestimation tools for Bayesian model averaging

[BMA] **Glossary**

For suggested citations, see the FAQ on citing Stata documentation.