Package ‘bbemkr’

February 19, 2015

Type Package

Title Bayesian bandwidth estimation for multivariate kernel regression with Gaussian error

Version 2.0

Date 2014-04-05

Depends R (>= 3.0.1), MASS

Author Han Lin Shang and Xibin Zhang

Maintainer Han Lin Shang <hanlin.shang@anu.edu.au>

Description Bayesian bandwidth estimation for Nadaraya-Watson type multivariate kernel regression with Gaussian error density

License GPL (>= 2)

LazyLoad yes

LazyData yes

URL https://sites.google.com/site/hanlinshangswebsite/

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-05 08:11:58

R topics documented:

`bbemkr-package` *Bayesian bandwidth estimation for multivariate kernel regression*

Description

Bayesian bandwidth estimation for Nadaraya-Watson type multivariate kernel regression with the Gaussian assumption of the error density and kernel-form error density

Details

This package designs for estimating bandwidths used in the Nadaraya-Watson kernel regression estimator. Assuming iid Gaussian error density that are uncorrelated to the regressors, the bandwidths are estimated using Markov chain Monte Carlo (MCMC) method, in particular by the random-walk Metropolis algorithm.

Author(s)

Han Lin Shang and Xibin Zhang

Maintainer: Han Lin Shang <H.Shang@soton.ac.uk>

References

H. L. Shang (2013) Bayesian bandwidth estimation for a semi-functional partial linear regression model with unknown error density, *Computational Statistics*, in press.

H. L. Shang (2013) Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density, *Computational Statistics and Data Analysis*, **67**, 185-198.

X. Zhang and M. L. King and H. L. Shang (2013) A sampling algorithm for bandwidth estimation in a nonparametric regression model with a flexible error density, http://www.buseco.monash.edu.au/ebs/pubs/wpapers/2011/wp 11.pdf.

X. Zhang and M. L. King and H. L. Shang (2013) Bayesian bandwidth selection for a nonparametric regression model with mixed types of regressors, working paper, http://www.buseco.monash.edu.au/ebs/pubs/wpapers/2013/v 13.pdf.

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

X. Zhang and M. L. King and R. J. Hyndman (2006) A Bayesian approach to bandwidth selection for multivariate kernel density estimation, *Computational Statistics and Data Analysis*, **50**, 3009-3031.

See Also

`np_gibbs`, `mcmcrecord_gaussian`, `warmup_gaussian`, `gibbs_admkr_nw`, `gibbs_admkr_erro`, `warmup_admkr`, `mcmcrecord_admkr`

`cost2_gaussian` *Negative of log posterior associated with the error variance*

Description

Calculates the negative of log posterior for the normal error variance, using the leave-one-out cross validated samples.

Usage

```
cost2_gaussian(x, data_x, data_y, prior_st)
```

Arguments

`x`	Log of square bandwidths
`data_x`	Regressors
`data_y`	Response variable
`prior_st`	Another tuning parameter of the prior of error variance, following inverse gamma distribution

Details

The prior of normal error variance follows an inverse-gamma distribution with hyperparameter `prior_st = 1`.

Value

Value of the cost function

Author(s)

Han Lin Shang

References

X. Zhang and R.D. Brooks and M.L. King (2009), A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

See Also

`np_gibbs`, `cost`

Examples

```
x = log(nrr(data_x, FALSE)^2)
cost2_gaussian(x, data_x = data_x, data_y = data_ynorm, prior_st = 1)
```

---

cost_admkr *Negative of log posterior associated with the bandwidths*

Description

Calculates the negative of log posterior, using the leave-one-out cross validated samples.

Usage

```
cost_admkr(x, data_x, data_y)
```

Arguments

`x`	Log of square bandwidths
`data_x`	Regressors
`data_y`	Response variable

Details

Bandwidth can be re-parameterized by a constant time optimal convergence rate, that is, $h = c * n^{rate}$.

Value

Value of the cost function

Author(s)

Han Lin Shang

References

H. L. Shang (2013) Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density, *Computational Statistics and Data Analysis*, **67**, 185-198.

X. Zhang, M. L. King and H. L. Shang (2013). A sampling algorithm for bandwidth estimation in a nonparametric regression model with a flexible error density. Working paper, http://users.monash.edu.au/~xzhang/zhang.king

X. Zhang, M. L. King and H. L. Shang (2013). Bayesian bandwidth selection for a nonparametric regression model with mixed types of regressors. Working paper, http://www.buseco.monash.edu.au/ebs/pubs/wpapers/2013/13.pdf

X. Zhang and M. L. King (2013). Gaussian kernel GARCH models. Working paper, http://users.monash.edu.au/~xzhang/zhar

See Also

gibbs_admkr_nw, gibbs_admkr_erro

Examples

```
x = log(c(nrr(data_x, FALSE),2)^2)
inicost = cost_admkr(x, data_x = data_x, data_y = data_ynorm)
```

---

cost_gaussian *Negative of log posterior associated with the bandwidths*

Description

Calculates the negative of log posterior, using the leave-one-out cross validated samples.

Usage

```
cost_gaussian(x, data_x, data_y, prior_p, prior_st)
```

Arguments

x	Log of square bandwidths
data_x	Regressors
data_y	Response variable
prior_p	A tuning parameter of the prior of error variance, following inverse gamma distribution
prior_st	Another tuning parameter of the prior of error variance, following inverse gamma distribution

Details

Bandwidth can be re-parameterized by a constant times optimal convergence rate, that is, $h = c * n^{rate}$. The prior of c^2 is assumed to follow an inverse-gamma prior with hyperparameters `prior_p = 2` and `prior_st = 1`.

Value

Value of the cost function

Author(s)

Han Lin Shang

References

X. Zhang and R.D. Brooks and M.L. King (2009), A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

See Also

`np_gibbs`, `cost2_gaussian`

Examples

```
x = log(nrr(data_x, FALSE)^2)
inicost = cost_gaussian(x, data_x = data_x, data_y = data_ynorm, prior_p = 2, prior_st = 1)
```

---

cov_chol — *Calculate log marginal likeliood from MCMC output*

Description

It is a type of candidate estimator for calculating log marginal likelihood, where the MCMC outputs are used for estimating posterior density.

Usage

```
cov_chol(xpost, data_x, data_y, alpha, prior_p, prior_st)
```

Arguments

`xpost`	MCMC output
`data_x`	Regressor
`data_y`	Response
`alpha`	Quantile of the critical value in calculating Geweke's log marginal likelihood
`prior_p`	Hyperparameter of the inverse-gamma prior
`prior_st`	Hyperparameter of inverse-gamma prior

Value

Log marginal likelihood

Author(s)

Han Lin Shang

References

J. Geweke (1998) Using simulation methods for Bayesian econometric models: inference, development, and communication, *Econometric Reviews*, **18**(1), 1-73.

See Also

`LaplaceMetropolis_gaussian`, `logdensity_gaussian`, `logpriors_gaussian`, `loglikelihood_gaussian`, `mcmcrecord_gaussian`

`cov_chol_admkr` *Calculate log marginal likelihood from MCMC output*

Description

It is a type of candidate estimator for calculating log marginal likelihood, where the MCMC outputs are used for estimating posterior density.

Usage

```
cov_chol_admkr(xpost, alpha, data_x, data_y)
```

Arguments

`xpost`	MCMC output
`alpha`	Quantile of the critical value in calculating Geweke's log marginal likelihood
`data_x`	Regressors
`data_y`	Response variable

Value

Log marginal likelihood

Author(s)

Han Lin Shang

References

J. Geweke (1998) Using simulation methods for Bayesian econometric models: inference, development, and communication, *Econometric Reviews*, **18**(1), 1-73.

See Also

LaplaceMetropolis_admkr, logdensity_admkr, logpriors_admkr, loglikelihood_admkr, mcmcrecord_admkr

---

data_x *Simulated three-dimensional regressors*

Description

Three-dimensional regressors are simulated from the uniform distribution by using `runif(1,0,1)`.

Usage

```
data(data_x)
```

Format

A data matrix with 100 observations and 3 x-variables.

Examples

```
nrr(data_x)
```

---

data_ynorm *Simulated response variable*

Description

The response variable is simulated from the functional form of $data_y norm = sin(2 * pi * x1) + 4 * (1 - x2) * (1 + x2) + 2 * x3/(1 + 0.8 * x3 * x3) + rnorm(1, 0, 0.9)$, where x1, x2 and x3 are simulated from a uniform distribution between 0 and 1.

Usage

```
data(data_ynorm)
```

Format

A data matrix of 100 by 1

Examples

```
data(data_ynorm)
```

`data_yt` *Simulated response variable*

Description

The response variable is simulated from the functional form of $data_y t = sin(2*pi*x1) + 4*(1-x2)*(1+x2) + 2*x3/(1+0.8*x3*x3) + rt(1, df = 4)$, where x1, x2 and x3 are simulated from a uniform distribution between 0 and 1.

Usage

```
data(data_yt)
```

Format

A data matrix of 100 by 1

Examples

```
data(data_yt)
```

`gibbs_admkr_erro` *Estimating bandwidth of the kernel-form error density*

Description

Implements the random-walk Metropolis algorithm to estimate the bandwidth of the kernel-form error density

Usage

```
gibbs_admkr_erro(xh, inicost, k, errorsizp, errorprob, data_x, data_y)
```

Arguments

xh	Log of square bandwidth in the kernel-form error density
inicost	Cost value
k	Iteration number
errorsizp	Step size of random-walk Metropolis algorithm
errorprob	Optimal convergence rate for drawing single or multiple parameters
data_x	Regressors
data_y	Response variable

Details

1) The log bandwidths of the regressors are initialized using the normal reference rule of Silverman (1986).

2) Conditioning on the variance parameter of the error density, we implement random-walk Metropolis algorithm to update the bandwidths, in order to achieve the minimum cost value.

3) The bandwidth of the kernel-form error density can be directly sampled.

4) Iterate steps 2) and 3) until the cost value is minimized.

5) Check the convergence of the parameters by examining the simulation inefficient factor (sif) value. The smaller the sif value is, the better convergence of the parameters is.

Value

`x`	Estimated bandwidth of the kernel-form error density
`cost`	Cost value, that is negative of log posterior
`accept_erro`	Accept or reject. `accept_erro = 1` indicates acceptance, while `accept_erro = 0` indicates rejection.
`errorsizp`	Step size of the random-walk Metropolis algorithm

Author(s)

Han Lin Shang

References

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

B. W. Silverman (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

See Also

`mcmcrecord_admkr`, `logdensity_admkr`, `loglikelihood_admkr`, `logpriors_admkr`, `gibbs_admkr_nw`

`gibbs_admkr_nw` *Estimating bandwidths of the regressors*

Description

Implements the random-walk Metropolis algorithm to estimate the bandwidths of the regressors

Usage

```
gibbs_admkr_nw(xh, inicost, k, mutsizp, prob, data_x, data_y)
```

Arguments

xh	Log of square bandwidths in the regression function
inicost	Cost value
k	Iteration number
mutsizp	Step size of random-walk Metropolis algorithm
prob	Optimal convergence rate for drawing single or multiple parameters
data_x	Regressors
data_y	Response variable

Details

1) The log bandwidths of the regressors are initialized using the normal reference rule of Silverman (1986).

2) Conditioning on the variance parameter of the error density, we implement random-walk Metropolis algorithm to update the bandwidths, in order to achieve the minimum cost value.

3) The bandwidth of the kernel-form error density can be directly sampled.

4) Iterate steps 2) and 3) until the cost value is minimized.

5) Check the convergence of the parameters by examining the simulation inefficient factor (sif) value. The smaller the sif value is, the better convergence of the parameters is.

Value

x	Estimated bandwidths of the regression function
cost	Cost value, that is negative of log posterior
accept_h	Accept or reject. accept_h=1 indicates acceptance, while accept_h=0 indicates rejection.
mutsizp	Step size of the random-walk Metropolis algorithm

Author(s)

Han Lin Shang

References

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

B. W. Silverman (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

See Also

mcmcrecord_admkr, logdensity_admkr, loglikelihood_admkr, logpriors_admkr, gibbs_admkr_erro ker *Type of kernel function*

Description

For data that have infinite support, Gaussian kernel is suggested. For data that have [-1, 1] support, other types of kernel can be used.

Usage

```
ker(u, kerntype = c("Gaussian", "Epanechnikov", "Quartic",
                    "Triweight", "Triangular", "Uniform"))
```

Arguments

u	A numeric object
kerntype	Type of kernel function

Details

Oftentimes, we deal with numeric values of infinite support, Gaussian kernel is commonly used. However, Epanechnikov kernel is the optimal kernel as measured by Mean Integrated Square Error. The difference among kernel functions is minor, but the influence of bandwidths is vital.

Value

Kernel value

Author(s)

Han Lin Shang

References

J. Fan and I. Gijbels (1996) Local Polynomial Modelling and Its Application. Chapman and Hall, London.

Q. Li and J. Racine (2007) Nonparametric Econometrics: Theory and Practice. Princeton University Press, New Jersey.

See Also

np_gibbs, gibbs_admkr_nw, gibbs_admkr_erro `kern` *Calculate the R square value and mean square error as measures of goodness of fit*

Description

To determine the goodness of fit, we calculate the R square value for leave-one-out cross validated regressors. To determine the googness of approximating regression function, we calculate the mean square error for leave-one-out cross validated regressors.

Usage

```
kern(h, data_x, data_y, xm)
```

Arguments

`h`	Bandwidth chosen after MCMC
`data_x`	Regressors
`data_y`	Response
`xm`	Values of true regression function

Value

Values of R square and mean square error

Author(s)

Han Lin Shang

See Also

`nrr`

`LaplaceMetropolis_admkr` *Laplace-Metropolis estimator of log marginal likelihood*

Description

As pointed out by Raftery (1996), the Laplace-Metropolis estimator performs well in calculating log marginal likelihood among other methods considered.

Usage

```
LaplaceMetropolis_admkr(theta, data_x, data_y,
    method = c("likelihood", "L1center", "median"))
```

Arguments

`theta`	MCMC output
`data_x`	Regressors
`data_y`	Response variable
`method`	Computing method. `L1center` and `median` are computationally fast

Details

The idea of the Laplace-Metropolis estimator is to avoid the limitations of the Laplace method by using posterior simulation to estimate the quantities it needs. The Laplace method for integrals is based on a Taylor series expansion of the real-valued function $f(u)$ of the d-dimensional vector u, and yields the approximation $P(D) \approx (2*pi)^(d/2)|A|^(1/2)P(D|\theta)P(\theta)$, where θ is the posterior mode of $h(\theta) = log(P(D|\theta)P(\theta))$, A is minus the inverse Hessian of $h(\theta)$ evaluated at $theta$, and d is the dimension of θ.

The simplest way to estimate θ from posterior simulation output, and probably the most accurate, is to compute $h(\theta^(t))$ for each $t = 1, \dots, T$ and take the value for which it is largest.

Value

Log marginal likelihood

Author(s)

Han Lin Shang

References

I. Ntzoufras (2009) Bayesian Modeling Using WinBUGS. John Wiley and Sons, Inc. New Jersey.

S. M. Lewis and A. E. Raftery (1997) Estimating Bayes factors via posterior simulation with the Laplace-Metropolis estimator, *Journal of the American Statistical Association*, **92**(438), 648-655.

A. E. Raftery (1996) Hypothesis testing and model selection, in Markov Chain Monte Carlo In Practice by W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Chapman and Hall, London.

See Also

`logdensity_admkr`, `logpriors_admkr`, `loglikelihood_admkr`, `mcmcrecord_admkr`

LaplaceMetropolis_gaussian

Laplace-Metropolis estimator of log marginal likelihood

Description

As pointed out by Raftery (1996), the Laplace-Metropolis estimator performs well in calculating log marginal likelihood among other methods considered.

Usage

```
LaplaceMetropolis_gaussian(theta, data = NULL, data_y, prior_p, prior_st,
        method = c("likelihood","L1center","median"))
```

Arguments

`theta`	MCMC output
`data`	Regressors
`data_y`	Response
`prior_p`	Hyperparameter of the inverse-gamma prior
`prior_st`	Hyperparameter of the inverse-gamma prior
`method`	Computing method. `L1center` and `median` are computationally fast

Details

The idea of the Laplace-Metropolis estimator is to avoid the limitations of the Laplace method by using posterior simulation to estimate the quantities it needs. The Laplace method for integrals is based on a Taylor series expansion of the real-valued function $f(u)$ of the d-dimensional vector u, and yields the approximation $P(D) \approx (2 * pi)^(d/2)|A|^(1/2)P(D|\theta)P(\theta)$, where θ is the posterior mode of $h(\theta) = log(P(D|\theta)P(\theta))$, A is minus the inverse Hessian of $h(\theta)$ evaluated at $theta$, and d is the dimension of θ.

The simplest way to estimate θ from posterior simulation output, and probably the most accurate, is to compute $h(\theta^(t))$ for each $t = 1, \ldots, T$ and take the value for which it is largest.

Value

Log marginal likelihood

Author(s)

Han Lin Shang

References

I. Ntzoufras (2009) Bayesian Modeling Using WinBUGS. John Wiley and Sons, Inc. New Jersey.

A. E. Raftery (1996) Hypothesis testing and model selection, in Markov Chain Monte Carlo In Practice by W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Chapman and Hall, London.

See Also

logdensity_gaussian, logpriors_gaussian, loglikelihood_gaussian, mcmcrecord_gaussian

---

logdensity_admkr — *Calculate an estimate of log posterior ordinate used in the log marginal density of Chib (1995).*

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
logdensity_admkr(tau2, cpost)
```

Arguments

tau2	Square of re-parameterized bandwidths and square of normal error variance
cpost	Simulation output of tau2 obtained from the MCMC iterations

Details

It should be noted that the posterior mode or maximum likelihood estimate can be computed from the simulation output at least approximately, if it is easy to evaluate the log-likelihood function for each draw in the simulation. Alternatively, one can make use of the posterior mean provided that there is no concern that it is a low density point.

Value

Value of the log density

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**(453), 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**(432), 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society. Series B*, **56**(1), 3-48.

See Also

logpriors_admkr, loglikelihood_admkr, mcmcrecord_admkr `logdensity_gaussian` *Calculate an estimate of log posterior ordinate used in the log marginal density of Chib (1995).*

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
logdensity_gaussian(tau2, cpost)
```

Arguments

`tau2`	Square of re-parameterized bandwidths and square of normal error variance
`cpost`	Simulation output of `tau2` obtained from the MCMC iterations

Details

It should be noted that the posterior mode or maximum likelihood estimate can be computed from the simulation output at least approximately, if it is easy to evaluate the log-likelihood function for each draw in the simulation. Alternatively, one can make use of the posterior mean provided that there is no concern that it is a low density point.

Value

Value of the log density

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

See Also

`logpriors_gaussian`, `loglikelihood_gaussian`, `mcmcrecord_gaussian` `loglikelihood_admkr` *Calculate the log likelihood used in the Chib's (1995) log marginal density*

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
loglikelihood_admkr(h2, data_x, data_y)
```

Arguments

`h2`	Square of re-parameterized bandwidths
`data_x`	Regressors
`data_y`	Response variable

Details

Calculates the log likelihood using the estimated averaged bandwidths of the regressors and estimated averaged variance of the error density

Value

The value of log likelihood, with parameters (bandwidths) estimated from the MCMC iterations

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

See Also

`logpriors_admkr`, `logdensity_admkr`, `mcmcrecord_admkr` loglikelihood_gaussian

Calculate the log likelihood used in the Chib's (1995) log marginal density

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
loglikelihood_gaussian(h2, data_x, data_y)
```

Arguments

h2	Square of re-parameterized bandwidths and square of normal error variance
data_x	Regressors
data_y	Response

Details

Calculates the log likelihood using the estimated averaged bandwidths of the regressors and estimated averaged variance of the error density

Value

The value of log likelihood, with parameters (bandwidths + normal error variance) estimated from the MCMC iterations

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

See Also

`logpriors_gaussian`, `logdensity_gaussian`, `mcmcrecord_gaussian` `logpriorh2` *Prior of square bandwidths*

Description

Prior of square bandwidths

Usage

```
logpriorh2(h2, prior_alpha = 1, prior_beta = 0.05)
```

Arguments

`h2`	Square bandwidths
`prior_alpha`	Hyperparameter of inverse-gamma prior
`prior_beta`	Hyperparameter of inverse-gamma prior

Details

Prior choice of bandwidths

Value

Prior value

Author(s)

Han Lin Shang

See Also

`mcmcrecord_gaussian`, `logdensity_gaussian`, `loglikelihood_gaussian`, `logpriors_gaussian`

`logpriors_admkr` *Calculate the log prior used in the log marginal density of Chib (1995).*

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
logpriors_admkr(h2, data_x)
```

Arguments

h2	Square of re-parameterized bandwidths
data_x	Regressors

Details

Calculate the log prior using the estimated averaged bandwidths of the regressors, obtained from the MCMC iterations

Value

Value of the log prior

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**(453), 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**(432), 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society. Series B*, **56**(1), 3-48.

See Also

logdensity_admkr, loglikelihood_admkr, mcmcrecord_admkr

---

logpriors_gaussian *Calculate the log prior used in the log marginal density of Chib (1995).*

Description

Log marginal likelihood = Log likelihood + Log prior - Log density

Usage

```
logpriors_gaussian(h2, data_x, prior_p, prior_st)
```

Arguments

h2	Square of re-parameterized bandwidths and square of normal error variance
data_x	Regressors
prior_p	Hyperparameter used in the inverse-gamma prior
prior_st	Hyperparameter used in the inverse-gamma prior

Details

Calculate the log prior using the estimated averaged bandwidths of the regressors and the estimated averaged variance of the error density, obtained from the MCMC iterations

Value

Value of the log prior

Author(s)

Han Lin Shang

References

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

See Also

`logdensity_gaussian`, `loglikelihood_gaussian`, `mcmcrecord_gaussian`

`mcmcrecord_admkr` *MCMC iterations*

Description

Estimated averaged bandwidths of the regressors of the kernel-form error density

Usage

```
mcmcrecord_admkr (x, inicost, mutsizp, errorsizp, warm = 100, M = 100, prob = 0.234,
    errorprob = 0.44, num_batch = 10, step = 10, data_x, data_y, xm, alpha = 0.05,
      mlike = c("Chib", "Geweke", "LaplaceMetropolis", "all"))
```

Arguments

`x`	Log of square bandwidth
`inicost`	Initial cost value
`mutsizp`	Step size of random-walk Metropolis algorithm. At each iteration, the value of `mutsizp` will alter depending on acceprance or rejection. As the number of iteration increases, the final acceptance probability will converge to the optimal rate, which is 0.234 for multiple parameters

`errorsizp`	Step size of random-walk Metropolis algorithm. At each iteration, the value of `errorsizp` will alter depending on acceprance or rejection. As the number of iteration increases, the final acceptance probability will converge to the optimal rate, which is 0.44 for single parameter
`warm`	Burn-in period
`M`	Number of MCMC iteration
`prob`	Optimal acceptance rate of random-walk Metropolis algorithm for the regression function
`errorprob`	Optimal acceptance rate of random-walk Metropolis algorithm for the error density
`num_batch`	Number of batch samples
`step`	Recording value at a specific step, in order to achieve iid samples and eliminate correlation
`data_x`	Regressors
`data_y`	Response variable
`xm`	Values of true regression function
`alpha`	Quantile of the critical value in calculating Geweke's log marginal likelihood
`mlike`	Method for calculating log marginal likelihood

Details

Akin to the burn-in period, it determines the retained bandwidths for the regressors and the variance of the error density for finite samples. It also calculates the simulation inefficient factor (SIF) value, R square, mean square error, and log marginal density by Chib (1995), Geweke (1999) and the Laplace Metropolis method describe in Raftery (1996).

Value

`sum_h`	Estimated parameters in an order of the bandwidths of the regressors, the variance parameter of the error density and cost value
`h2`	Estimated parameters in an order of the square bandwidths of the regressors, the square variance parameter of the error density
`sif`	Simulation inefficient factor. The small it is, the better the method is in general
`mutsizp`	Step size of random-walk Metropolis algroithm for each iteration of `MCMCrecord`
`cpost`	Simulation output of square bandwidths obtained from MCMC
`ghost`	Simulation output of square bandwidths obtained from MCMC
`accept_nw`	Acceptance rate of random-walk Metropolis algorithm for the regression function
`accept_erro`	Acceptance rate of random-walk Metropolis algorithm for the kernel-form error density
`marginalike`	Log marginal likelihood
`R2`	R square
`MSE`	Mean square error

Note

Time-consuming for large iterations.

Author(s)

Han Lin Shang

References

H. L. Shang (2013) Bayesian bandwidth estimation for a semi-functional partial linear regression model with unknown error density, *Computational Statistics*, in press.

H. L. Shang (2013) Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density, *Computational Statistics and Data Analysis*, **67**, 185-198.

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

J. Geweke (1998) Using simulation methods for Bayesian econometric models: inference, development, and communication, *Econometric Reviews*, **18**(1), 1-73.

A. E. Raftery (1996) Hypothesis testing and model selection, in Markov Chain Monte Carlo In Practice by W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Chapman and Hall, London.

See Also

`logdensity_admkr`, `logpriors_admkr`, `loglikelihood_admkr`, `warmup_admkr`

`mcmcrecord_gaussian` *MCMC iterations*

Description

Estimated averaged bandwidths of the regressors and averaged variance parameter of the normal error density

Usage

```
mcmcrecord_gaussian(x, inicost, mutsizp, warm = 100, M = 100, prob = 0.234,
          num_batch = 10, step = 10, data_x, data_y, xm,
          alpha = 0.05, prior_p = 2, prior_st = 1,
          mlike = c("Chib", "Geweke", "LaplaceMetropolis", "all"))
```

Arguments

`x`	Log of square bandwidth
`inicost`	Initial cost value
`mutsizp`	Step size of random-walk Metropolis algorithm. At each iteration, the value of `mutsizp` will alter depending on acceprance or rejection. As the number of iteration increases, the final acceptance probability will converge to the optimal rate, which is 0.234 for multiple parameters
`warm`	Burn-in period
`M`	Number of MCMC iteration
`prob`	Optimal acceptance rate of random-walk Metropolis algorithm
`num_batch`	Number of batch samples
`step`	Recording value at a specific step, in order to achieve iid samples and eliminate correlation
`data_x`	Regressors
`data_y`	Response variable
`xm`	Values of true regression function
`alpha`	Quantile of the critical value in calculating Geweke's log marginal likelihood
`prior_p`	Hyperparameter of inverse-gamma prior
`prior_st`	Hyperparameter of inverse-gamma prior
`mlike`	Method for calculating log marginal likelihood

Details

Akin to the burn-in period, it determines the retained bandwidths for the regressors and the variance of the error density for finite samples. It also calculates the simulation inefficient factor (SIF) value, R square, mean square error, and log marginal density by Chib (1995), Geweke (1999) and the Laplace Metropolis method describe in Raftery (1996).

Value

`sum_h`	Estimated parameters in an order of the bandwidths of the regressors, the variance parameter of the error density and cost value
`h2`	Estimated parameters in an order of the square bandwidths of the regressors, the square variance parameter of the error density
`sif`	Simulation inefficient factor. The small it is, the better the method is in general
`mutsizp`	Step size of random-walk Metropolis algroithm for each iteration of `MCMCrecord`
`cpost`	Simulation output of square bandwidths and square normal error variance obtained from MCMC
`accept`	Acceptance rate of random-walk Metropolis algorithm
`marginalike`	Log marginal likelihood
`R2`	R square
`MSE`	Mean square error

Note

Time-consuming for large iterations.

Author(s)

Han Lin Shang

References

H. L. Shang (2013) Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density, *Computational Statistics and Data Analysis*, **67**, 185-198.

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

S. Chib and I. Jeliazkov (2001) Marginal likelihood from the Metropolis-Hastings output, *Journal of the American Statistical Association*, **96**, 453, 270-281.

S. Chib (1995) Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association*, **90**, 432, 1313-1321.

M. A. Newton and A. E. Raftery (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion), *Journal of the Royal Statistical Society*, **56**, 3-48.

J. Geweke (1998) Using simulation methods for Bayesian econometric models: inference, development, and communication, *Econometric Reviews*, **18**(1), 1-73.

A. E. Raftery (1996) Hypothesis testing and model selection, in Markov Chain Monte Carlo In Practice by W. R. Gilks, S. Richardson and D. J. Spiegelhalter, Chapman and Hall, London.

See Also

`logdensity_gaussian`, `logpriors_gaussian`, `loglikelihood_gaussian`, `warmup_gaussian`

`NadarayaWatsonkernel` *Nadaraya-Watson kernel estimator*

Description

Nadaraya (1964) and Watson (1964) proposed to estimate `m` as a locally weighted average, using a kernel as a weighting function.

Usage

```
NadarayaWatsonkernel(x, y, h, gridpoint)
```

Arguments

`x`	A set of `x` observations.
`y`	A set of `y` observations.
`h`	Optimal bandwidth chosen by the user.
`gridpoint`	A set of gridpoints.

Details

$\frac{\sum_{i=1}^{n} K_h(x-x_i)y_i}{\sum_{j=1}^{n} K_h(x-x_j)}$, where K is a kernel function with a bandwidth `h`.

Value

`gridpoint`	A set of gridpoints.
`mh`	Density values corresponding to the set of gridpoints.

Author(s)

Han Lin Shang

References

M. Rosenblatt (1956) Remarks on some nonparametric estimates of a density function, *The Annals of Mathematical Statistics*, **27**(3), 832-837.

E. Parzen (1962) On estimation of a probability density function and mode, *The Annals of Mathematical Statistics*, **33**(3), 1065-1076.

E. A. Nadaraya (1964) On estimating regression, *Theory of probability and its applications*, **9**(1), 141-142.

G. S. Watson (1964) Smooth regression analysis, *Sankhya: The Indian Journal of Statistics (Series A)*, **26**(4), 359-372.

Examples

```
x = rnorm(100)
y = rnorm(100)
NadarayaWatsonkernel(x, y, h = 2, gridpoint = seq(-3, 3, length.out = 100))
```

`np_gibbs` *Estimating bandwidths of the regressors*

Description

Implements the random-walk Metropolis algorithm to estimate the bandwidths of the regressors

Usage

```
np_gibbs(xh, inicost, k, mutsizp, prob, data_x, data_y, prior_p, prior_st)
```

Arguments

`xh`	Log of square bandwidths
`inicost`	Cost value
`k`	Iteration number
`mutsizp`	Step size of random-walk Metropolis algorithm
`prob`	Optimal covergence rate
`data_x`	Regressors
`data_y`	Response variable
`prior_p`	Hyperparameter used in the inverse-gamma prior
`prior_st`	Hyperparameter used in the inverse-gamma prior

Details

1) The log bandwidths of the regressors are initialized using the normal reference rule of Silverman (1986).

2) Conditioning on the variance parameter of the error density, we implement random-walk Metropolis algorithm to update the bandwidths, in order to achieve the minimum cost value.

3) The variance of the error density can be directly sampled.

4) Iterate steps 2) and 3) until the cost value is minimized.

5) Check the convergence of the parameters by examining the simulation inefficient factor (sif) value. The smaller the sif value is, the better convergence of the parameters is.

Value

`x`	Estimated bandwidths of the regression function
`sigma2`	Estimated variance of the normal error density
`cost`	Cost value
`accept_h`	Accept or reject. `accept_h=1` indicates acceptance, while `accept_h=0` indicates rejection.
`mutsizp`	Step size of random-walk Metropolis

Author(s)

Han Lin Shang

References

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

B. W. Silverman (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

See Also

`mcmcrecord_gaussian`, `logdensity_gaussian`, `loglikelihood_gaussian`, `logpriors_gaussian`

---

nrr — *Normal reference rule for estimating bandwidths*

Description

The simplest method for estimating bandwidths, based on normal density assumption.

Usage

```
nrr(data_x, logband = TRUE)
```

Arguments

`data_x`	Regressors
`logband`	When `logband=TRUE`, log bandwidths are given. When `logband=FALSE`, bandwidths are given.

Value

Bandwidths.

Author(s)

Han Lin Shang

References

X. Zhang and R. D. Brooks and M. L. King (2009) A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation, *Journal of Econometrics*, **153**, 21-32.

A. W. Bowman and A. Azzalini (1997) Applied smoothing techniques for data analysis, Oxford University Press, London.

D. W. Scott (1992) Multivariate density estimation: theory, practice, and visualization, John Wiley and Sons, New York.

B. W. Silverman (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

Examples

```
nrr(data_x, logband = FALSE)
```

---

warmup_admkr *Burn-in period*

Description

By minimising the cost value, the function estimates the bandwidths of the regressors and kernel-form error density for the burn-in period

Usage

```
warmup_admkr(x, inicost, mutsizp, errorsizp, warm = 100, prob = 0.234,
    errorprob = 0.44, data_x, data_y)
```

Arguments

`x`	Log of square bandwidths
`inicost`	Cost value
`mutsizp`	Step size of random-walk Metropolis algorithm for the regressors
`errorsizp`	Step size of random-walk Metropolis algorithm for the kernel-form error density
`warm`	Number of burn-in iterations
`prob`	Optimal covergence rate of random-walk Metropolis algorithm for the regressors
`errorprob`	Optimal covergence rate of random-walk Metropolis algorithm for the kernel-form error density
`data_x`	Regressors
`data_y`	Response variable

Value

x	Log of square bandwidths
cost	Cost value
mutsizp	Step size of random-walk Metropolis algorithm for the regressors
errorsizp	Step size of random-walk Metropolis algorithm for the kernel-form error density

Author(s)

Han Lin Shang

See Also

mcmcrecord_admkr, logdensity_admkr, loglikelihood_admkr, logpriors_admkr

---

warmup_gaussian *Burn-in period*

Description

By minimizing the cost value, the function estimates the bandwidths of the regressors and normal error variance parameter for the burn-in period

Usage

```
warmup_gaussian(x, inicost, mutsizp, warm = 100, prob = 0.234, data_x, data_y,
       prior_p = 2, prior_st = 1)
```

Arguments

x	Log of square bandwidths
inicost	Cost value
mutsizp	Step size of random-walk Metropolis algorithm
warm	Number of burn-in iterations
prob	Optimal covergence rate of random-walk Metropolis algorithm
data_x	Regressors
data_y	Response variable
prior_p	Hyperparameter of the inverse-gamma prior
prior_st	Hyperparameter of the inverse-gamma prior

Value

`x`	Log of square bandwidths
`sigma2`	Estimate of normal error variance
`cost`	Cost value
`mutsizplast`	Final step size of random-walk Metropolis algorithm
`mutsizp`	Step size of random-walk Metropolis algorithm

Author(s)

Han Lin Shang

See Also

`mcmcrecord_gaussian`, `logdensity_gaussian`, `loglikelihood_gaussian`, `logpriors_gaussian`

`xm` *Values of true regression function*

Description

The realization of regression function

Usage

```
data(xm)
```

Format

A data matrix of 100 by 1

Examples

```
data(xm)
```

Index