Package 'ZVCV'

January 24, 2019

Type Package

Title Zero-Variance Control Variates

Version 1.0.0

Date 2018-12-20

Description Zero-variance control variates (ZV-CV, Mira et al. (2013) <doi:10.1007/s11222-012-9344-6>) is a post-processing method to reduce the variance of Monte Carlo estimators of expectations using the derivatives of the log target. Once the derivatives are available, the only additional computational effort is in solving a linear regression problem. Recently, this method has been extended to higher dimensions using regularisation (South et al., 2018 <arXiv:1811.05073>). This package can be used to easily perform ZV-CV or regularised ZV-CV when a set of samples, derivatives and function evaluations are available. Additional functions for applying ZV-CV to two estimators for the normalising constant of the posterior distribution in Bayesian statistics are also supplied.

License GPL (>= 2)

LazyLoad yes

Imports Rcpp (>= 0.11.0), glmnet, abind, mvtnorm, partitions, stats

LinkingTo Rcpp, RcppArmadillo

LazyData true

Encoding UTF-8

RoxygenNote 6.1.1

NeedsCompilation yes

Author Leah F. South [aut, cre] (<https://orcid.org/0000-0002-5646-2963>)

Maintainer Leah F. South <leah.south@hdr.qut.edu.au>

Repository CRAN

Date/Publication 2019-01-24 00:00:02 UTC

R topics documented:

`evidence` *Evidence estimation with ZV-CV*

Description

The function `evidence_CTI` uses ZV-CV on the controlled thermodynamic integration estimator for the normalising constant.

Usage

```
evidence_CTI(samples, loglike, der_loglike, der_logprior, temperatures,
  temperatures_all, most_recent, obs_estim_choose, obs_estim, options)

evidence_SMC(samples, loglike, der_loglike, der_logprior, temperatures,
  temperatures_all, most_recent, obs_estim_choose, obs_estim, options)
```

Arguments

`samples`	An N by d by T matrix of samples from the T power posteriors, where N is the number of samples and d is the dimension of the target distribution
`loglike`	An N by T matrix of log likelihood values corresponding to `samples`
`der_loglike`	An N by d by T matrix of the derivatives of the log likelihood with respect to the parameters, with parameter values corresponding to `samples`
`der_logprior`	An N by d by T matrix of the derivatives of the log prior with respect to the parameters, with parameter values corresponding to `samples`
`temperatures`	A vector of length T of temperatures for the power posterior temperatures
`temperatures_all`	An adjusted vector of length tau of temperatures. Better performance should be obtained with a more conservative temperature schedule. See `Expand_Temperatures` for a function to adjust the temperatures.
`most_recent`	A vector of length tau which gives the indices in the original temperatures that the new temperatures correspond to.
`obs_estim_choose`	See `zvcv`.
`obs_estim`	See `zvcv`.
`options`	See `zvcv`.

Value

The function `evidence_CTI` returns a list, containing the following components:

- `log_evidence_PS1`: The 1st order quadrature estimate for the log normalising constant
- `log_evidence_PS2`: The 2nd order quadrature estimate for the log normalising constant
- `regression_LL`: The set of tau `zvcv` type returns for the 1st order quadrature expectations
- `regression_vLL`: The set of tau `zvcv` type returns for the 2nd order quadrature expectations

The function `evidence_SMC` returns a list, containing the following components:

- `log_evidence`: The logged SMC estimate for the normalising constant
- `regression_SMC`: The set of tau `zvcv` type returns for the expectations

Author(s)

Leah F. South

References

Mira, A., Solgi, R., & Imparato, D. (2013). Zero variance Markov chain Monte Carlo for Bayesian estimators. Statistics and Computing, 23(5), 653-662.

South, L. F., Oates, C. J., Mira, A., & Drovandi, C. (2019). Regularised zero variance control variates for high-dimensional variance reduction. https://arxiv.org/abs/1811.05073

See Also

See `Expand_Temperatures` for a function that can be used to find stricter (or less stricter) temperature schedules based on the conditional effective sample size. See an example at `VDP` and see ZVCV for more package details.

`Expand_Temperatures` *Adjusting the temperature schedule*

Description

This function is used to adjust the temperature schedule so that it is more (or less) strict than the original.

Usage

```
Expand_Temperatures(temperatures, loglike, rho,
  bisec_tol = .Machine$double.eps^0.25)
```

Arguments

`temperatures`	A vector of length T temperatures for the power posterior temperatures.
`loglike`	An N by T matrix of log likelihood values corresponding to `samples`.
`rho`	The tolerance for the new temperatures, which are selected so that the CESS at each temperature is $\rho * N$ where N is the population size.
`bisec_tol`	The tolerance for the bisection method used in selecting temperatures. The default is `.Machine$double.eps^0.25`

Value

A list is returned, containing the following components:

- `temperatures_all`: The new set of temperatures of length tau.
- `relevant_samples`: A vector of length tau containinng indices to show which particle sets the new temperatures are based on.
- `logw`: An N by tau matrix of log normalised weights of the particles

Author(s)

Leah F. South

References

South, L. F., Oates, C. J., Mira, A., & Drovandi, C. (2019). Regularised zero variance control variates for high-dimensional variance reduction. https://arxiv.org/abs/1811.05073

See Also

See `evidence` for functions to estimate the evidence, `VDP` for an example and ZVCV for more package details.

`getX` *The function* `getX` *is used to get the matrix of covariates for the regression based on a specified polynomial order.*

Description

The function `getX` is used to get the matrix of covariates for the regression based on a specified polynomial order.

Usage

```
getX(samples, derivatives, polyorder)
```

Arguments

`samples`	An N by d matrix of samples from the target
`derivatives`	An N by d matrix of derivatives of the log target with respect to the parameters
`polyorder`	The order of the polynomial.

Value

The design matrix for the regression.

`helper_functions` *Useful helper functions*

Description

The function `logsumexp` is used for stable computation of log(sum(exp(x))), which is useful when summing weights for example.

Usage

```
logsumexp(x)
```

Arguments

`x`	The values for which you want to compute log(sum(exp(x)))

Value

The stable result of log(sum(exp(x)))

See Also

See ZVCV for more package details.

VDP *Example of estimation using SMC*

Description

This example illustrates how ZV-CV can be used for post-processing of results from SMC. In particular, we use ZV-CV to estimate posterior expectations and the evidence for a single SMC run of this example based on the Van der Pol oscillatory differential equations (Van der Pol, 1926). Further details about this example and applications to ZV-CV can be found in Oates et al. (2017) and South et al. (2019).

Given that the focus of this R package is on ZV-CV, we assume that samples have already been obtained from SMC and put into the correct format. One could use the R package `RcppSMC` or implement their own sampler in order to obtain results like this. The key is to make sure the derivatives of the log likelihood and log prior are stored, along with the inverse temperatures.

Usage

```
data(VDP)
```

Format

A list containing the following :

N The size of the SMC population

rho The tolerance for the new temperatures, which are selected so that the CESS at each temperature is $\rho * N$ where N is the population size.

temperatures A vector of length T of inverse power posterior temperatures

samples An N by d by T matrix of samples from the T power posteriors, where d is the dimension of the target distribution. The samples are transformed to be on the log scale and all derivatives are with respect to log samples.

loglike An N by T matrix of log likelihood values corresponding to `samples`

logprior An N by T matrix of log prior values corresponding to `samples`

der_loglike An N by d by T matrix of the derivatives of the log likelihood with respect to the parameters, with parameter values corresponding to `samples`

der_logprior An N by d by T matrix of the derivatives of the log prior with respect to the parameters, with parameter values corresponding to `samples`

References

Oates, C. J., Girolami, M. & Chopin, N. (2017). Control functionals for Monte Carlo integration. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79(3), 695-718.

South, L. F., Oates, C. J., Mira, A., & Drovandi, C. (2019). Regularised zero-variance control variates for high-dimensional variance reduction.

Van der Pol, B. (1926). On relaxation-oscillations. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, 2(11), 978-992.

See Also

See ZVCV for more package details.

Examples

```
set.seed(1)

# Load the SMC results
data(VDP)

# Set up the list of control variates to choose from
options <- list()
# Vanilla Monte Carlo
options[[1]] <- list(polyorder = 0)
# Standard ZV-CV with polynomial order selected through cross-validation
options[[2]] <- list(polyorder = Inf, regul_reg = FALSE)

##############################
# Posterior expectation - The true expectation is 0.9852 to 4 decimal places
##############################

# Note the exp() because samples and derivatives were stored on the log scale
# but we are interested in the expectation on the original scale
posterior <- zvcv(exp(VDP$samples[,,8]), VDP$samples[,,8],
VDP$der_loglike[,,8] + VDP$der_logprior[,,8], options = options)
posterior$expectation # The posterior expectation estimate
posterior$polyorder # The selected polynomial order

##############################
# Evidence estimation - The true logged evidence is 10.36 to 2 decimal places
##############################

## Not run:
# Getting additional temperatures based on maintaing a CESS of 0.99N rather than 0.9N
temp <- Expand_Temperatures(VDP$temperatures, VDP$loglike, 0.99)
VDP$temperatures_new <- temp$temperatures_all # the new temperatures
VDP$most_recent <- temp$relevant_samples # the samples associated with the new temperatures

## End(Not run)



# Evidence estimation using the SMC identity
Z_SMC <- evidence_SMC(VDP$samples, VDP$loglike, VDP$der_loglike, VDP$der_logprior,
VDP$temperatures, VDP$temperatures_new, VDP$most_recent, options = options)
Z_SMC$log_evidence

# Evidence estimation using the CTI identity
Z_CTI <- evidence_CTI(VDP$samples, VDP$loglike, VDP$der_loglike, VDP$der_logprior,
VDP$temperatures, VDP$temperatures_new, VDP$most_recent, options = options)
Z_CTI$log_evidence_PS2
```

zvcv *ZV-CV for general expectations*

Description

The function zvcv is used to perform (regularised) ZV-CV given a set of samples, derivatives and function evaluations.

Usage

```
zvcv(integrand, samples, derivatives, log_weight,
  integrand_logged = FALSE, obs_estim_choose, obs_estim,
  options = list(polyorder = 2, regul_reg = TRUE, alpha_elnet = 1, nfolds
  = 10, apriori = (1:NCOL(samples)), intercept = TRUE, polyorder_max =
  Inf), folds_choose = 5)
```

Arguments

integrand	The integrand, i.e. a set of N evaluations of the function of interest.
samples	An N by d matrix of samples from the target
derivatives	An N by d matrix of derivatives of the log target with respect to the parameters
log_weight	(optional) A vector of length N containing log weights of the samples. The default is equal weights.
integrand_logged	(optional) Sometimes it is better to input the integrand on the logged scale for stability. If the actual integrand is the exponential of integrand, then integrand_logged = TRUE. Otherwise, the default of integrand_logged = FALSE should be used.
obs_estim_choose	(optional) The indices of the samples to be used in estimating the MSE for cross validation. This is not relevant if options is a single list of specifications. If options is a list of multiple control variate specifications, then the default is to use 10 percent of the full data set for estimating the MSE.
obs_estim	(optional) The indices of the samples to be used in evaluating the integrand. If this is missing or NULL then the full data set is used for both estimating the polynomial and evaluating the integrand. Otherwise, the samples from obs_estim are used in evaluating the integral and the remainder are used in estimating the polynomial.
options	A list of control variate specifications. This can be a single list containing the elements below (the defaults are used for elements which are not specified). Alternatively, it can be a list of lists containing any or all of the elements below. Where the latter is used, the function zvcv automatically selects the best performing option based on cross-validation.

- polyorder: The order of the polynomial, with a default of 2. A value of Inf will get the cross-validation method to choose between orders.

- regul_reg: A flag for whether regularised regression is to be used. The default is TRUE, i.e. regularised regression is used.
- alpha_elnet: The alpha parameter for elastic net. The default is 1, which correponds to LASSO. A value of 0 would correspond to ridge regression.
- nfolds: The number of folds used in cross-validation to select lambda for LASSO or elastic net. The default is 10.
- apriori: The indices of the parameters to use in the polynomial. The default is to use all parameters $1:d$ where d is the dimension of the target. If only the first and third derivatives should be used, then this would be specified by using apriori = c(1,3) (alternatively, this can be done by only using the relevant columns in samples and derivatives).
- intercept: A flag for whether the intercept should be estimated or fixed to the empirical mean of the integrand in the estimation set. The default is to include an intercept (intercept = TRUE) as this tends to lead to better variance reductions. Note that an intercept = TRUE flag may be changed to intercept = FALSE within the function if integrand_logged = TRUE and a NaN is encountered. See South et al. (2018) for further details.
- polyorder_max: The maximum allowable polynomial order. This may be used to prevent memory issues in the case that the polynomial order is selected automatically. You will be prompted to see if you wish to continue if the selected polynomial order results in a design matrix with more than ten million elements. A default maximum polynomial order will be selected if the polyorder is infinite and in this case a warning will be given. Recall that setting your default R settings to options(warn=1) will ensure that you receive these warnings in real time. Optimal polynomial order selection may go to at most this maximum value, or it may stop earlier.

folds_choose — The number of folds used in k-fold cross-validation for selecting the optimal control variate. Depending on the options, this may include selection of the optimal polynomial order, regression type and subset of parameters in the polynomial. The default is five.

Value

A list is returned, containing the following components:

- expectation: The estimate of the expectation.
- num_select: The number of non-zero coefficients in the polynomial.
- mse: The mean square error for the evaluation set.
- integrand_logged: The integrand_logged input stored for reference.
- obs_estim: The obs_estim input stored for reference.
- polyorder: The polyorder value used in the final estimate.
- regul_reg: The regul_reg flag used in the final estimate.
- alpha_elnet: The alpha_elnet value used in the final estimate.
- nfolds: The nfolds value used in the final estimate.
- apriori: The apriori value used in the final estimate.
- intercept: The intercept flag used in the final estimate.

Author(s)

Leah F. South

References

Mira, A., Solgi, R., & Imparato, D. (2013). Zero variance Markov chain Monte Carlo for Bayesian estimators. Statistics and Computing, 23(5), 653-662.

South, L. F., Oates, C. J., Mira, A., & Drovandi, C. (2019). Regularised zero variance control variates for high-dimensional variance reduction. https://arxiv.org/abs/1811.05073

See Also

See evidence for functions which use zvcv to estimate the normalising constant of the posterior. See an example at VDP and see ZVCV for more package details.

Examples

```
# Estimating the mean of theta1 when theta is bivariate normally distributed with:
mymean <- c(1,2)
mycov <- matrix(c(1,0.5,0.5,2),nrow=2)

# Perfect draws from the target distribution (could be replaced with
# approximate draws from e.g. MCMC or SMC)
N <- 50
require(mvtnorm)
set.seed(1)
samples <- rmvnorm(N, mean = mymean, sigma = mycov)
integrand <- samples[,1]

# derivatives of Gaussian wrt x
derivatives <- t( apply(samples,1,function(x) -solve(mycov)%*%(x - mymean)) )

# Estimates without ZV-CV (i.e. vanilla Monte Carlo integration)
# Without the ZVCV package
mean(integrand)
# With the ZVCV package
zvcv(integrand,samples,derivatives,options = list(polyorder = 0))$expectation

# ZV-CV with fixed specifications
# 2nd order polynomial with LASSO
sprintf("%.15f",zvcv(integrand, samples, derivatives)$expectation)
# 2nd order polynomial with OLS
sprintf("%.15f",zvcv(integrand, samples, derivatives,
options = list(polyorder = 2, regul_reg = FALSE))$expectation)

# ZV-CV with cross validation
# Choose between OLS and LASSO, with the order selected using cross validation
myopts <- list(list(polyorder = Inf, regul_reg = FALSE),list(polyorder = Inf))
temp <- zvcv(integrand,samples,derivatives,options = myopts)
temp$polyorder # The chosen control variate order
sprintf("%.15f",temp$expectation) # The expectation based on the minimum MSE control variate
```

```
temp$regul_reg # Flag for if the chosen control variate uses regularisation
```

ZVCV_package *Zero-Variance Control Variates*

Description

Zero-variance control variates (ZV-CV, Mira et al. (2013) <doi:10.1007/s11222-012-9344-6>) is a post-processing method to reduce the variance of Monte Carlo estimators of expectations using the derivatives of the log target. Once the derivatives are available, the only additional computational effort is in solving a linear regression problem. Recently, this method has been extended to higher dimensions using regularisation (South et al., 2018). This package can be used to easily perform ZV-CV or regularised ZV-CV when a set of samples, derivatives and function evaluations are available. Additional functions for applying ZV-CV to two estimators for the normalising constant of the posterior distribution (also known as the evidence) are also supplied.

The main functionality is available through

- zvcv: The main function to estimate expectations using (regularised) ZV-CV. This function uses a set of N samples along with the associated derivatives of the log target and evaluations of the function of interest.
- evidence_CTI: A function to estimate the evidence using the controlled thermodynamic integration (CTI) identity, with samples from a set of T power posteriors.
- evidence_SMC: A function to estimate the evidence using ZV-CV on the SMC evidence identity, with samples from a set of T power posteriors.

Some additional helper functions are available. See Expand_Temperatures and helper_functions for further details.

An example of estimation posterior expectations and the evidence from SMC samples is available at VDP.

Author(s)

Leah F. South

References

Mira, A., Solgi, R., & Imparato, D. (2013). Zero variance Markov chain Monte Carlo for Bayesian estimators. Statistics and Computing, 23(5), 653-662.

South, L. F., Oates, C. J., Mira, A., & Drovandi, C. (2018). Regularised zero variance control variates. https://arxiv.org/abs/1811.05073

Index