# Package: iwmm (via r-universe)

June 16, 2024

**Title** Importance weighted moment matching

**Version** 0.0.1

**Description** iwmm provides functions for adaptive importance sampling.

**License** GPL (>=3)

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Depends** R (>= 3.1.2)

**Imports** checkmate, matrixStats (>= 0.52), posterior (>= 1.5.0), stats

**Suggests** bayesplot, knitr, parallel, rmarkdown, rstan, testthat

**Enhances** brms, cmdstanr

**VignetteBuilder** knitr

**Config/testthat/parallel** false

**Config/testthat/edition** 3

**Repository** https://topipa.r-universe.dev

**RemoteUrl** https://github.com/topipa/iwmm

**RemoteRef** HEAD

**RemoteSha** b40fdc398f4d8ffe5e32aa2a8367cb3d3db6607c

## Contents


constrain_draws . . . . . 2
example_iwmm_model . . . . . 2
log_prob_draws . . . . . 3
moment_match . . . . . 3
moment_match.brmsfit . . . . . 5
moment_match.CmdStanFit . . . . . 6
moment_match.stanfit . . . . . 7
shift . . . . . 8

shift_and_cov . . . . . 8
shift_and_scale . . . . . 9
unconstrain_draws . . . . . 9

**Index** **10**

---

`constrain_draws` *Constrain all draws from a fitted model*

---

## Description

Constrain all draws from a fitted model

## Usage

```
constrain_draws(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | model fit object |
| `...` | arguments passed to methods |

## Value

constrained draws

---

`example_iwmm_model` *Example Stan model for importance weighted moment matching*

---

## Description

Provides example models (with data) that are ready for use with IWMM.

## Usage

```
example_iwmm_model(model = "normal_model")
```

## Arguments

| | |
|---|---|
| `model` | Character specifying which model code to return. Currently "normal_model" is implemented. |

## Value

List containing model code and corresponding data.

## log_prob_draws *Return log probability of posterior*

### Description

Return log probability of posterior

### Usage

```
log_prob_draws(fit, ...)
```

### Arguments

| | |
|---|---|
| fit | model fit object |
| ... | arguments passed to methods |

### Value

TODO

## moment_match *Generic importance weighted moment matching algorithm.*

### Description

Generic importance weighted moment matching algorithm.

Generic importance weighted moment matching algorithm for matrices.

### Usage

```
moment_match(x, ...)

## S3 method for class 'matrix'
moment_match(
  x,
  log_prob_prop_fun,
  log_prob_target_fun = NULL,
  log_ratio_fun = NULL,
  expectation_fun = NULL,
  log_expectation_fun = FALSE,
  draws_transformation_fun = NULL,
  is_method = "psis",
  adaptation_method = "iwmm",
  k_threshold = 0.5,
  cov_transform = TRUE,
```

```
  split = FALSE,
  restart_transform = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | A matrix of draws. Must be unconstrained. |
| ... | Further arguments passed to log_prob_prop_fun, log_prob_target_fun and log_ratio_fun. |
| log_prob_prop_fun | Log density of the proposal. The function takes argument draws. |
| log_prob_target_fun | Log density of the target for importance sampling. The function takes argument draws. |
| log_ratio_fun | Log of the density ratio for importance sampling (target/proposal). The function takes argument draws. |
| expectation_fun | Optional argument, NULL by default. A function whose expectation is being computed. The function takes arguments draws. |
| log_expectation_fun | Logical indicating whether the expectation_fun returns its values as logarithms or not. Defaults to FALSE. If set to TRUE, the expectation function must be non-negative (before taking the logarithm). Ignored if expectation_fun is NULL. |
| draws_transformation_fun | Optional argument, NULL by default. A function that transforms draws before computing expectation. The function takes arguments draws. |
| is_method | Which importance sampling method to use. Currently only psis is supported. |
| adaptation_method | Which adaptation method to use. Currently only iwmm is supported. |
| k_threshold | Threshold value for Pareto k values above which the moment matching algorithm is used. The default value is 0.5. |
| cov_transform | Logical; Indicates whether to match the covariance of the samples or not. If FALSE, only the mean and marginal variances are matched. Default is TRUE. |
| split | Logical; Indicate whether to do the split transformation or not at the end of moment matching. FALSE by default. |
| restart_transform | Logical; When split is TRUE, indicates whether to start the second transformation from the original model parameters or the transformed parameters. If split is FALSE, this is ignored. |

## Value

Returns a list with: transformed draws, updated importance weights, and the pareto k diagnostic value. If expectation_fun is given, also returns the expectation.

moment_match.brmsfit *Generic importance weighted moment matching algorithm for* brmsfit *objects. See additional arguments from* moment_match.matrix

## Description

Generic importance weighted moment matching algorithm for brmsfit objects. See additional arguments from moment_match.matrix

## Usage

```
## S3 method for class 'brmsfit'
moment_match(
  x,
  log_prob_target_fun = NULL,
  log_ratio_fun = NULL,
  target_observation_weights = NULL,
  expectation_fun = NULL,
  log_expectation_fun = FALSE,
  constrain = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | A fitted brmsfit object. |
| log_prob_target_fun | Log density of the target. The function takes argument draws, which are the unconstrained draws. Can also take the argument fit which is the stan model fit. |
| log_ratio_fun | Log of the density ratio (target/proposal). The function takes argument draws, which are the unconstrained draws. Can also take the argument fit which is the stan model fit. |
| target_observation_weights | A vector of weights for observations for defining the target distribution. A value 0 means dropping the observation, a value 1 means including the observation similarly as in the current data, and a value 2 means including the observation twice. |
| expectation_fun | Optional argument, NULL by default. A function whose expectation is being computed. The function takes arguments draws. |
| log_expectation_fun | Logical indicating whether the expectation_fun returns its values as logarithms or not. Defaults to FALSE. If set to TRUE, the expectation function must be non-negative (before taking the logarithm). Ignored if expectation_fun is NULL. |

| | |
|---|---|
| constrain | Logical specifying whether to return draws on the constrained space? Default is TRUE. |
| ... | Further arguments passed to moment_match.matrix. |

## Value

Returns a list with 3 elements: transformed draws, updated importance weights, and the pareto k diagnostic value. If expectation_fun is given, also returns the expectation.

---

moment_match.CmdStanFit

*Generic importance weighted moment matching algorithm for* CmdStanFit *objects. See additional arguments from* moment_match.matrix

---

## Description

Generic importance weighted moment matching algorithm for CmdStanFit objects. See additional arguments from moment_match.matrix

## Usage

```
## S3 method for class 'CmdStanFit'
moment_match(
  x,
  log_prob_target_fun = NULL,
  log_ratio_fun = NULL,
  constrain_draws = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | A fitted CmdStanFit object. |
| log_prob_target_fun | Log density of the target. The function takes argument draws, which are the unconstrained draws. |
| log_ratio_fun | Log of the density ratio (target/proposal). The function takes argument draws, which are the unconstrained draws. |
| constrain_draws | Logical specifying whether to return draws on the constrained space. Draws are also constrained for computing expectations. Default is TRUE. |
| ... | Further arguments passed to moment_match.matrix. |

## Value

Returns a list with 3 elements: transformed draws, updated importance weights, and the pareto k diagnostic value.

moment_match.stanfit *Generic importance weighted moment matching algorithm for* stanfit *objects. See additional arguments from* moment_match.matrix

## Description

Generic importance weighted moment matching algorithm for stanfit objects. See additional arguments from moment_match.matrix

## Usage

```
## S3 method for class 'stanfit'
moment_match(
  x,
  log_prob_target_fun = NULL,
  log_ratio_fun = NULL,
  target_observation_weights = NULL,
  expectation_fun = NULL,
  log_expectation_fun = FALSE,
  constrain_draws = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | A fitted stanfit object. |
| log_prob_target_fun | Log density of the target. The function takes argument draws, which are the unconstrained draws. Can also take the argument fit which is the stan model fit. |
| log_ratio_fun | Log of the density ratio (target/proposal). The function takes argument draws, which are the unconstrained draws. Can also take the argument fit which is the stan model fit. |
| target_observation_weights | A vector of weights for observations for defining the target distribution. A value 0 means dropping the observation, a value 1 means including the observation similarly as in the current data, and a value 2 means including the observation twice. |
| expectation_fun | Optional argument, NULL by default. A function whose expectation is being computed. The function takes arguments draws. |
| log_expectation_fun | Logical indicating whether the expectation_fun returns its values as logarithms or not. Defaults to FALSE. If set to TRUE, the expectation function must be non-negative (before taking the logarithm). Ignored if expectation_fun is NULL. |

constrain_draws
: Logical specifying whether to return draws on the constrained space. Draws are also constrained for computing expectations. Default is TRUE.

... : Further arguments passed to `moment_match.matrix`.

## Value

Returns a list with 3 elements: transformed draws, updated importance weights, and the pareto k diagnostic value. If expectation_fun is given, also returns the expectation.

---

# shift *Shift a matrix of draws to their weighted mean.*

---

## Description

Shift a matrix of draws to their weighted mean.

## Usage

```
shift(draws, lw)
```

## Arguments

| | |
|---|---|
| draws | A matrix of draws. |
| lw | A vector representing the log-weight of each draw. |

## Value

List with the shift that was performed, and the new draws matrix.

---

# shift_and_cov *Shift a matrix of draws to their weighted mean and scale the covariance to match the weighted covariance.*

---

## Description

Shift a matrix of draws to their weighted mean and scale the covariance to match the weighted covariance.

## Usage

```
shift_and_cov(draws, lw)
```

## Arguments

| | |
|---|---|
| draws | A matrix of draws. |
| lw | A vector representing the log-weight of each draw. |

### Value

List with the shift and mapping that were performed, and the new draws matrix.

---

`shift_and_scale` *Shift a matrix of draws to their weighted mean and scale the marginal variances to match the weighted marginal variances.*

---

### Description

Shift a matrix of draws to their weighted mean and scale the marginal variances to match the weighted marginal variances.

### Usage

```
shift_and_scale(draws, lw)
```

### Arguments

| | |
|---|---|
| `draws` | A matrix of draws. |
| `lw` | A vector representing the log-weight of each draw. |

### Value

List with the shift and scaling that were performed, and the new draws matrix.

---

`unconstrain_draws` *Unconstrain all draws from a fitted model*

---

### Description

Unconstrain all draws from a fitted model

### Usage

```
unconstrain_draws(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | model fit object |
| `...` | arguments passed to methods |

### Value

unconstrained draws

# Index

constrain_draws, 2

example_iwmm_model, 2

log_prob_draws, 3

moment_match, 3
moment_match.brmsfit, 5
moment_match.CmdStanFit, 6
moment_match.stanfit, 7

shift, 8
shift_and_cov, 8
shift_and_scale, 9

unconstrain_draws, 9