

Package: dentist (via r-universe)

October 29, 2024

Title Dents the Likelihood Surface to Estimate Parameter Uncertainty

Version 0.0.4

Description This will sample points around a specified distance from the maximum likelihood estimates. This should be a better way to estimate uncertainty than using the Hessian of the likelihood equation. It works by ``denting" the likelihood surface to make a ridge at the desired difference in log likelihood and then ``walks" around this dented surface, sampling points.

License GPL (>= 3)

URL <https://bomeara.github.io/dentist/>,
<https://github.com/bomeara/dentist>

BugReports <https://github.com/bomeara/dentist/issues>

Imports graphics, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository <https://phylotastic.r-universe.dev>

RemoteUrl <https://github.com/bomeara/dentist>

RemoteRef HEAD

RemoteSha 44f0c3eee2873ad14563a68ab27603f7bdeeb1f1

Contents

dent_likelihood	2
dent_propose	3
dent_walk	3
plot.dentist	6

print.dentist	7
quiet_fn	7
sphere_propose	8
summary.dentist	8

Index	10
--------------	-----------

dent_likelihood	<i>Dents the likelihood surface</i>
-----------------	-------------------------------------

Description

This takes any values that are better (lower) than the desired negative log likelihood and reflects them across the best_neglnL + delta line, "denting" the likelihood surface.

Usage

```
dent_likelihood(neglnL, best_neglnL, delta = 2)
```

Arguments

neglnL	The original negative log likelihood
best_neglnL	The negative log likelihood at the optimum; other values will be greater than this.
delta	How far from the optimal negative log likelihood to focus samples

Value

The transformed negative log likelihood

Examples

```
sims <- stats::rnorm(100, mean=17)
possible_means <- seq(from=16, to=18, length.out=100)
results_normal <- rep(NA, length(possible_means))
results_dented <- results_normal
dnorm_to_run <- function(par, sims) {
  return(-sum(dnorm(x=sims, mean=par, log=TRUE)))
}
best_neglnL <- optimize(dnorm_to_run,interval=range(possible_means),
  sims=sims, maximum=FALSE)$objective
for (i in seq_along(possible_means)) {
  results_normal[i] <- dnorm_to_run(possible_means[i], sims=sims)
  results_dented[i] <- dent_likelihood(results_normal[i], best_neglnL=best_neglnL, delta=2)
}

plot(possible_means, results_normal)
lines(possible_means, results_dented)
```

dent_propose	<i>Propose new values This proposes new values using a normal distribution centered on the original parameter values, with desired standard deviation. If any proposed values are outside the bounds, it will propose again.</i>
--------------	--

Description

Propose new values This proposes new values using a normal distribution centered on the original parameter values, with desired standard deviation. If any proposed values are outside the bounds, it will propose again.

Usage

```
dent_propose(old_params, lower_bound = -Inf, upper_bound = Inf, sd = 1)
```

Arguments

old_params	The original parameter values
lower_bound	Minimum parameter values to try. One for all or a vector of the length of par.
upper_bound	Maximum parameter values to try. One for all or a vector of the length of par.
sd	Standard deviation to use for the proposals. One for all or a vector of the length of par.

Value

A vector of the new parameter values

dent_walk	<i>Sample points from along a ridge</i>
-----------	---

Description

This "dents" the likelihood surface by reflecting points better than a threshold back across the threshold (think of taking a hollow plastic model of a mountain and punching the top so it's a volcano). It then uses essentially a Metropolis-Hastings walk to wander around the new rim. It adjusts the proposal width so that it samples points around the desired likelihood. This is better than using the curvature at the maximum likelihood estimate since it can actually sample points in case the assumptions of the curvature method do not hold. It is better than varying one parameter at a time while holding others constant because that could miss ridges: if I am fitting $5=x+y$, and get a point estimate of (3,2), the reality is that there are an infinite range of values of x and y that will sum to 5, but if I hold x constant it looks like y is estimated very precisely. Of course, one could just fully embrace the Metropolis-Hastings lifestyle and use a full Bayesian approach.

Usage

```
dent_walk(
  par,
  fn,
  best_neglnL,
  confidence_level = 0.95,
  delta = NULL,
  nsteps = 1000,
  print_freq = 50,
  lower_bound = 0,
  upper_bound = Inf,
  adjust_width_interval = 100,
  badval = 1e+09,
  sd_vector = NULL,
  debug = FALSE,
  restart_after = 50,
  quiet = TRUE,
  sphere_probability = 0,
  ...
)
```

Arguments

par	Starting parameter vector, generally at the optimum. If named, the vector names are used to label output parameters.
fn	The likelihood function, assumed to return negative log likelihoods
best_neglnL	The negative log likelihood at the optimum; other values will be greater than this.
confidence_level	The confidence level represents the long-run proportion of CIs (at the given confidence level) that theoretically contain the true value of the parameter. For example, out of all intervals computed at the 95% level, 95% of them should contain the parameter's true value
delta	How far from the optimal negative log likelihood to focus samples
nsteps	How many steps to take in the analysis
print_freq	Output progress every print_freq steps.
lower_bound	Minimum parameter values to try. One for all or a vector of the length of par.
upper_bound	Maximum parameter values to try. One for all or a vector of the length of par.
adjust_width_interval	When to try automatically adjusting proposal widths
badval	Bad negative log likelihood to return if a non-finite likelihood is returned
sd_vector	Vector of the standard deviations to use for proposals. Generated automatically if NULL
debug	If TRUE, prints out much more information during a run

restart_after	Sometimes the search can get stuck outside the good region but still accept moves. After this many steps without being inside the good region, restart from one of the past good points
quiet	If TRUE, will quiet the likelihood function outputs
sphere_probability	The probability of using a sphere proposal near the bounds instead of the regular proposal
...	Other arguments to fn.

Details

While running, it will display the current range of likelihoods in the desired range (by default, the best negative log likelihood + 2 negative log likelihood units) and the parameter values falling in that range. If things are working well, the range of values will stabilize during a search.

The algorithm tunes: if it is moving too far away from the desired likelihoods, it will decrease the proposal width; if it staying in areas better than the desired likelihood, it will increase the proposal width. It will also expand the proposal width for parameters where the extreme values still appear good enough to try to find out the full range for these values.

In general, the idea of this is not to give you a pleasingly narrow range of possible values – it is to try to find the actual uncertainty, including finding any ridges that would not be seen in univariate space.

Value

A dentist object containing results, the data.frame of negative log likelihoods and the parameters associated with them; acceptances, the vector of whether a proposed move was accepted each step; best_neglnL, the best value passed into the analysis; delta, the desired offset; all_ranges, a summary of the results.

Examples

```
# Univariate case
sims <- stats::rnorm(100, mean=17)
possible_means <- seq(from=16, to=18, length.out=100) # for optimize

# Make sure we have a function that takes in a parameters vector, other arguments if needed,
# and returns the negative log likelihood
dnorm_to_run <- function(par, sims) {
  return(-sum(stats::dnorm(x=sims, mean=par, log=TRUE)))
}

optimized_results <- stats::optimize(dnorm_to_run,interval=range(possible_means),
  sims=sims, maximum=FALSE)
best_par <- optimized_results$minimum
names(best_par) <- "mean"
best_neglnL <- optimized_results$objective

dented_results <- dent_walk(par=best_par, fn=dnorm_to_run, best_neglnL=best_neglnL, sims=sims)
plot(dented_results)
```

```
# Multivariate case
sims <- stats::rlnorm(100, meanlog=1, sdlog=3)

dlnorm_to_run <- function(par, sims) {
  return(-sum(stats::dlnorm(sims, meanlog=par[1], sdlog=par[2], log=TRUE)))
}

optimized_results <- stats::optim(c(meanlog=.9, sdlog=2.9), dlnorm_to_run, sims=sims)
best_par <- optimized_results$par
best_neglnL <- optimized_results$value

dented_results <- dent_walk(par=best_par, fn=dlnorm_to_run, best_neglnL=best_neglnL, sims=sims)
plot(dented_results)
```

plot.dentist	<i>Plot the dented samples This will show the univariate plots of the parameter values versus the likelihood as well as bivariate plots of pairs of parameters to look for ridges.</i>
--------------	--

Description

Plot the dented samples This will show the univariate plots of the parameter values versus the likelihood as well as bivariate plots of pairs of parameters to look for ridges.

Usage

```
## S3 method for class 'dentist'
plot(x, local.only = FALSE, binary_color = TRUE, ...)
```

Arguments

x	An object of class dentist
local.only	Boolean indicating whether to trim x and y lims to be near accepted points
binary_color	Boolean indicating whether to color points by inside or outside a threshold or by likelihood (only do latter with two free parameters)
...	Other arguments to pass to plot

print.dentist	<i>Print dentist print summary of output from dent_walk</i>
---------------	---

Description

Print dentist print summary of output from dent_walk

Usage

```
## S3 method for class 'dentist'  
print(x, ...)
```

Arguments

x	An object of class dentist
...	Other arguments (not used)

quiet_fn	<i>Quiet internal functions</i>
----------	---------------------------------

Description

Quiet internal functions

Usage

```
quiet_fn(x)
```

Arguments

x	A function to be quieted
---	--------------------------

sphere_propose	<i>Propose new values based on points near bounds This proposes new values based on wiggling a little bit away from the values near the bounds of the good region</i>
----------------	---

Description

Propose new values based on points near bounds This proposes new values based on wiggling a little bit away from the values near the bounds of the good region

Usage

```
sphere_propose(
  results,
  delta,
  old_params,
  lower_bound = -Inf,
  upper_bound = Inf,
  sd = 1
)
```

Arguments

results	Data.frame of results so far
delta	The maximum distance from the best value to consider
old_params	The original parameter values
lower_bound	Minimum parameter values to try. One for all or a vector of the length of par.
upper_bound	Maximum parameter values to try. One for all or a vector of the length of par.
sd	Standard deviation to use for the proposals. One for all or a vector of the length of par.

Value

A vector of the new parameter values

summary.dentist	<i>Summarize dentist Display summary of output from dent_walk</i>
-----------------	---

Description

Summarize dentist Display summary of output from dent_walk

Usage

```
## S3 method for class 'dentist'  
summary(object, ...)
```

Arguments

<code>object</code>	An object of class <code>dentist</code>
<code>...</code>	Other arguments (not used)

Index

dent_likelihood, [2](#)
dent_propose, [3](#)
dent_walk, [3](#)

plot.dentist, [6](#)
print.dentist, [7](#)

quiet_fn, [7](#)

sphere_propose, [8](#)
summary.dentist, [8](#)