Package 'pomdpSolve'

February 8, 2022

Title Interface to 'pomdp-solve' for Partially Observable Markov Decision Processes

Version 1.0.0

Date 2022-02-04

Description Installs 'pomdp-solve', a program to solve Partially Observable Markov Decision Processes (POMDPs) using a variety of exact and approximate value iteration algorithms. Smallwood and Sondik (1973) <doi:10.1287/opre.21.5.1071>.

Classification/ACM G.4, G.1.6, I.2.6

URL https://github.com/mhahsler/pomdp

BugReports https://github.com/mhahsler/pomdpSolve/issues

Depends R (>= 3.5.0)

Encoding UTF-8

License GPL (>= 3)

Copyright pomdp-solve is Copyright (C) Anthony R. Cassandra; LASPack is Copyright (C) Tomas Skalicky; lp-solve is Copyright (C) Michel Berkelaar, Kjell Eikland, Peter Notebaert; all other code is Copyright (C) Michael Hahsler.

RoxygenNote 7.1.2.9000

NeedsCompilation yes

Author Michael Hahsler [aut, cph, cre], Anthony R. Cassandra [ctb, cph]

Maintainer Michael Hahsler <mhahsler@lyle.smu.edu>

Repository CRAN

Date/Publication 2022-02-08 16:00:02 UTC

R topics documented:

find_pomdp_solve 2

Index **4**

`find_pomdp_solve` *Find the executable for 'pomdp-solve'*

Description

This package installs the executable for 'pomdp-solve' to solve Partially Observable Decision Processes (POMDPs). We provide a function to find the installed executable.

Usage

```
find_pomdp_solve()
```

Details

Note that this package only provides a direct interface to the executable. A more convenient and powerful interface is provided by the function `pomdp::solve_POMDP()` in package **pomdp**.

The executable of pomdp-solve in this direct interface needs to be called with `system2()` and creates files with the value function and the policy graph.

Value Function

The value function is returned as files with the extension `.alpha` in the format:

```
A
V1 V2 V3 ... VN

A
V1 V2 V3 ... VN

...
```

Where `A` is an action number and the `V1` through `VN` are real values representing the components of a particular vector that has the associated action. The action number is the 0-based index of the action as specified in the input POMDP file. The vector represents the coefficients of a hyperplane representing one facet of the piecewise linear and convex (PWLC) value function. Note that the length of the lists needs to be equal to the number of states in the POMDP.

Policy Graph

The policy graph is returned as a file with the extension `.pg`. Each line of the file represents one node of the policy graph and its contents are:

```
N A  Z1 Z2 Z3 ...
...
```

Here `N` is a node ID giving the node a unique name, numbered sequentially and lining up sequentially with the value function vectors in the corresponding output `.alpha` file above.

The `A` is the action number defined for this node; it is an integer referring to the the POMDP file actions by its 0-based index number. These are followed by a list of node IDs, one for each

observation. Thus the list will have a length equal to the number of observations in the POMDP. This list specifies the transitions in the policy graph. The nth number in the list will be the index of the node that follows this one when the observation received is n.

Details about the file formats and pomdp-solve can be found in the References section.

Value

`find_pomdp_solve()` returns the path to the 'pomdp-solve' executable as a string or stops with an error.

References

Anthony R. Cassandra, pomdp-solve documentation, https://www.pomdp.org/code/index.html

Examples

```
# find the location of the pomdp-solve executable
find_pomdp_solve()

# get pomdp-solve options
system2(find_pomdp_solve(), args = "-h")

# solve a POMDP file that ships with this package in a temporay directory
tmp <- tempdir()
pomdp_file <- file.path(tmp, "tiger.aaai.POMDP")
file.copy(system.file("tiger.aaai.POMDP", package = "pomdpSolve"), pomdp_file)

system2(find_pomdp_solve(), args = paste("-pomdp", pomdp_file, "-method grid"))
dir(tmp)

# read the raw policy graph
read.table(file.path(tmp, "tiger.aaai-0.pg"))

# read the raw value function
readLines(file.path(tmp, "tiger.aaai-0.alpha"))
```

Index

find_pomdp_solve, 2

pomdp-solve *(*find_pomdp_solve*)*, 2
pomdp::solve_POMDP(), *2*
pomdpsolve *(*find_pomdp_solve*)*, 2

system2(), *2*