

University of Massachusetts Lowell

Electrical & Computer Engineering Department

CMINDS Research Center

Machine-Learning Program
Usage Summary

Author

H. S. Al-Dayaa

Revision

1.0

Date

April 18, 2006

REVISION HISTORY

Revision	Date	Editor	Description
1.0	04/18/2006	H.S.A.	Initial revision.

MLPROG USAGE SUMMARY

The Machine-Learning Program (MLProg) is stand-alone robust software utility we developed to implement a number of reinforcement learning techniques, specifically the one we developed for our research. MLProg offers the user a lot of flexibility via a large number of user-defined configuration and parameters, plentiful details about the different steps of the applied reinforcement learning process logged in a text file called the Process Information Log file, and agent path data logged in text files called the Agent Physical Path Data files for the physical paths and the Agent Hypothetical Path Data files for the learning paths.

Figure 1 illustrates a high-level flowchart of MLProg and its input and output interfaces.

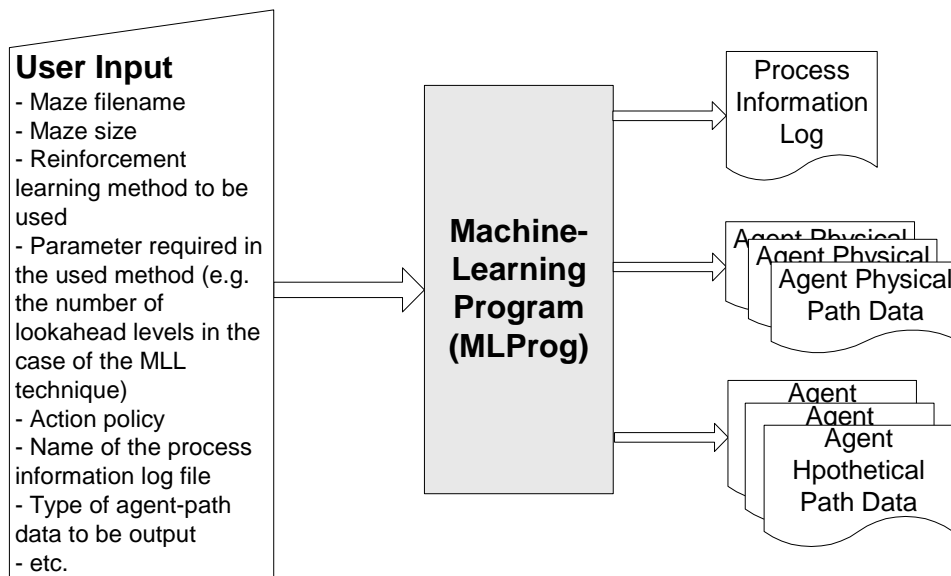


Figure 1 – The Machine-Learning Program and its Interfaces

Table 1 illustrates the Help Table of MLProg which shows the different options given to the user to select from.

Usage: mlprog <options>	
Options:	
-i <file>	Filename of input maze matrix
-x <integer>	Maze X size (number of columns in maze matrix)
-y <integer>	Maze Y size (number of rows in maze matrix)
-o <file>	Filename of agent output log
-p <integer>	Number of physical paths
-l <integer>	Number of learning paths per physical path
[-m <integer>]	Method (default=0)
	0: Learning with Q-value only
	1: Learning with Distance-to-Goal Only
	2: Learning with Q-value and Frequency (need w1 & w3)
	3: Learning with Distance and Frequency (w2 & w3)
	4: Learning with Q-value and Distance (need w1 & w3)
	5: Learning with Q-value, Distance, and Frequency (need w1, w2, w3)
	10: Physical with Distance only
	11: Physical with Distance and Frequency
	12: Physical with Distance and Frequency using Closed-Loop Omitting
	13: Physical with Distance and Frequency using Loop Pinching
[-n <integer>]	Number of lookahead levels N (default=1)
[-c <file>]	File where controlled first-path actions are
[-f <integer>]	Path-file mode (default=0)
	0: No path files
	1: Physical path files only
	2: Learning path files only (including the lrn num and R/Q rwrdr files)
	3: Physical and learning path files (including the lrn num and R/Q rwrdr files)
[-v <integer>]	Level of verbose in agent output log file
[-e]	Initial full-exploration flag
[-g]	Physical-Greedy flag
[-r]	Learning-Greedy flag
[-A <float>]	Step-size parameter ALPHA (default=0.100)
[-G <float>]	Discount rate GAMMA (default=0.900)
[-L <float>]	Lookahead weight factor LAMBDA (default=0.500)
[-T <integer>]	Initial Boltzmann constant TAW (default=50000)
[-1 <float>]	Q-value weight w1 (default=0.000)
[-2 <float>]	Distance-to-Goal weight w2 (default=0.000)
[-3 <float>]	Visited-Number weight w3 (default=0.000)
[-a]	Print information about mlprog
[-h]	Print this Help Table

Table 1 – Help Table of the Machine-Learning Program

The Process Information Log file mainly includes copies of the environment as recorded in the agent brain after each path and detailed physical and hypothetical path information. Depending on the verbose level specified by the user, the Agent Information Log file can include many details such as intermediate and final values of the different rewards. The higher the verbose level is, the more detailed information is logged in the Agent Information Log file.

The Agent Physical Path Data files and the Agent Hypothetical Path Data files consist of lists of the $[x, y]$ coordinates of the different agent states in each corresponding path. These files can be used with any graphic tool (e.g. Matlab) to display the agent path, plot the number of learning steps per each learning path, etc.

MLPROG PACKAGE

The following is the list of files delivered in the MLProg package:

- mlprog_win: The MLProg program executable built for Windows OS.
- mlprog_sol: The MLProg program executable built for Solaris OS.
- mlprog_lin: The MLProg program executable built for Linux OS.
- maze1_8x6.in: An example of the 'input maze matrix' file.
- cntrl1_maze1_8x6.in: An example of the 'controlled first-path actions' file.
- MLProg_Usage_Summary_rev1_0.pdf: The Machine-Learning Program Usage Summary document (this document).

REFERENCES

- [1] Al-Dayaa, H. S., “Theorization, Implementation, System Architecture, and Analysis of Fast Reinforcement Learning Techniques, with Application to Autonomous Agents”, doctorate dissertation, *University of Massachusetts Lowell*, USA, 2006.