RL agent Donor agnostic no penalty no u donors table

Table linear basis function as value function

format shortG

seed = 0

No U donors

allmodelfoldincs = readmatrix("allmodelfoldincs.txt")

modelinitcons = readmatrix("modelinitcons.txt")

rawfoldincs = readmatrix("rawfoldincs.txt")

rawinitcons = readmatrix("rawinitcons.txt")

truecon = readtable("true con.xlsx");

truecon(:,1)=[];

rawdonornames = truecon.Properties.VariableNames;

rawdonornames = string(rawdonornames);

rawdonornames = strrep(rawdonornames,"\_","-")

% rawdonornames = ["T054","T046","T051","T062","T031","T031","T052","T038","T036","T036","T036","T066","T066","T066","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA4","UA141","UA141","UA141","UA141","UA141","UA6","UA6","UA6","UA6","UA6"]

donorlist = ["T054","T046","T051","T062","T031","T052","T038","T036","T066","UA4","UA141","UA6"]

donoridxmap = {1,2,3,4,5:6,7,8,9:11,12:14,15:27,28:32,33:37}

Run training and validation

Randomization:

-6:3 training:validation split for tall the T donors

u donors never included

numdonors = 6

All validation set printed

initializedatasets(seed,allmodelfoldincs,modelinitcons,rawfoldincs,rawinitcons,rawdonornames,donoridxmap,donorlist);

numdonors = 6;

qagent = qinittraining(seed,numdonors);

save("agent\_2481"+seed,"qagent")

rawfoldincs(:,15:end) = []

rawinitcons(15:end) = []

% dqnagent = load("agent\_248\_hailmary"+seed,"dqnagent").dqnagent

[simulationPlots,simulationResults] = simwithset(rawfoldincs,rawinitcons,rawdonornames,qagent);

[rewardMatrix,totalRewards,totalReward,averageReward] = calculatescore(6,simulationResults)

writematrix(rewardMatrix,"rewardMatrix\_2481"+seed+".xlsx")

writematrix(totalRewards,"totalRewards\_2481"+seed+".xlsx")

function [trialfoldincs, trialinitcons, trialdonornames] = initializedatasets(seed,allmodelfoldincs,modelinitcons,rawfoldincs,rawinitcons,rawdonornames,donoridxmap,donorlist)

rng(seed);

trainingvector = sort(randperm(9,6),"ascend");

trainingset = donorlist(trainingvector)

validationvector = [];

for i=1:12

if ismember(i,trainingvector)==false

validationvector = [validationvector i];

end

end

validationvector = sort(validationvector,"ascend");

validationset = donorlist(validationvector)

availmodelfoldincs = [allmodelfoldincs(:,1:numel(modelinitcons)\*9)];

trialmodelfoldincs = [];

for i=trainingvector

trialmodelfoldincs = [trialmodelfoldincs availmodelfoldincs(:,(numel(modelinitcons)\*(i-1)+1):(numel(modelinitcons)\*(i-1)+numel(modelinitcons)))];

end

writematrix(trialmodelfoldincs, "modelfoldincs.txt")

% trialfoldincs = [];

% trialinitcons = [];

% trialdonornames = [];

% for i=validationvector

% trialfoldincs = [trialfoldincs rawfoldincs(:,donoridxmap{i})];

% trialinitcons = [trialinitcons rawinitcons(donoridxmap{i})];

% % POTENTIALLY BETTER IF ALL NORMALIZED TO 500000

% trialdonornames = [trialdonornames rawdonornames(donoridxmap{i})];

% end

end

function qagent = qinittraining(seed,numdonors)

rng(0);

% create first environment

env = modelEnv241;

env.useCustom = false;

env.numdonors = numdonors;

obs = getObservationInfo(env);

acts = getActionInfo(env);

validateEnvironment(env)

% hyperparameters

discountFactor = 1;

epsilon = 1;

epsilondecay = 0.00125;

Xs = @(obs,acts) tileBsFcn2(obs,acts);

W0 = zeros(1080,1);

qfunctionoptions = rlRepresentationOptions;

% qfunction = rlQValueRepresentation(qtable,obs,acts)

qfunction = rlQValueRepresentation({Xs,W0},obs,acts,qfunctionoptions);

qagentoptions = rlQAgentOptions;

qagentoptions.DiscountFactor = discountFactor;

qagentoptions.EpsilonGreedyExploration.Epsilon = epsilon;

qagent = rlQAgent(qfunction,qagentoptions);

trainopts = rlTrainingOptions;

trainopts.MaxEpisodes = 1500;

trainopts.StopTrainingCriteria = "EpisodeCount";

trainopts.StopTrainingValue = 1500;

% trainopts.StopTrainingCriteria = "AverageReward";

% trainopts.StopTrainingValue = 15;

%trainopts.Plots = 'none';

%trainopts.Verbose = true;

plot(env)

results = train(qagent,env,trainopts);

end

function [plots,experiences] = simwithset(foldExMaps,day3cons,donornames,sarsaagent3)

plots = cell(size(foldExMaps,2),1);

experiences = cell(size(foldExMaps,2),1);

for i=1:size(foldExMaps,2)

%creating foldexmap

foldExMap1 = foldExMaps(:,i);

foldExMap1 = foldExMap1(~isnan(foldExMap1));

idx = foldExMap1==1;

idx(1) = false;

foldExMap1(idx) = [];

%creating new env

env3 = modelEnv241;

env3.useCustom = true;

env3.foldExMap = foldExMap1;

%NORMALIZED INIT CONS

%

env3.initialState = [100000/1000000,1];

env3.modelInfo = donornames(i)

validateEnvironment(env3)

% plot now

plot(env3);

experiences{i} = sim(sarsaagent3,env3);

plots{i} = env3.Figure;

end

end

function [rewardMatrix,totalRewards,totalReward,averageReward] = calculatescore(maxSeriesLength,simulationResults)

maxSeriesLength = 5;

rewardMatrix = [0:(maxSeriesLength-1)]';

for i=1:size(simulationResults,1)

seriesRewards = simulationResults{i}.Reward.Data;

if size(seriesRewards,1)<maxSeriesLength

numZeros = maxSeriesLength - size(seriesRewards,1);

for j=1:numZeros

seriesRewards = [seriesRewards; 0];

end

end

rewardMatrix = [rewardMatrix seriesRewards];

end

totalRewards = [];

for i=2:size(rewardMatrix,2)

totalRewards = [totalRewards sum(rewardMatrix(:,i))];

end

totalReward = sum(totalRewards);

averageReward = mean(totalRewards);

end

function Xex = tileBsFcn2(obs,acts)

actions = [0.18 0.21 0.24 0.27 0.3 1];

numactions = numel(actions);

bounds1 = 0:500000:18000000;

bounds2 = 0:500000:18000000;

Xex = zeros(((numel(bounds1)-1)\*3+(numel(bounds2)-1)\*2),numactions);

for i=1:(numactions)

% day 3

for a=1:(numel(bounds1)-1)

Xex(a,i) = all([obs(1)>bounds1(a);obs(1)<=bounds1(a+1);obs(2)==1;acts(1)==(actions(i))]);

end

% day 5

for a=1:(numel(bounds1)-1)

Xex(a+(numel(bounds1)-1),i) = all([obs(1)>bounds1(a);obs(1)<=bounds1(a+1);obs(2)==2;acts(1)==(actions(i))]);

end

% day 7

for a=1:(numel(bounds1)-1)

Xex(a+(numel(bounds1)-1)\*2,i) = all([obs(1)>bounds1(a);obs(1)<=bounds1(a+1);obs(2)==3;acts(1)==(actions(i))]);

end

% day 9

for b=1:(numel(bounds2)-1)

Xex(b+(numel(bounds1)-1)\*3,i) = all([obs(1)>bounds2(b);obs(1)<=bounds2(b+1);obs(2)==4;acts(1)==(actions(i))]);

end

% day 11

for b=1:(numel(bounds2)-1)

Xex(b+(numel(bounds1)-1)\*3+(numel(bounds2)-1),i) = all([obs(1)>bounds2(b);obs(1)<=bounds2(b+1);obs(2)==5;acts(1)==(actions(i))]);

end

end

Xex=reshape(Xex,[numel(Xex),1]);

end