junedmunshi

How to implement epsilon greedy strategy / policy

Filed under: Uncategorized — 4 Comments
March 30, 2012
Epsilon greedy policy is a way of selecting random actions with uniform distribution from a set of available actions.
Using this policy either we can select random action with epsilon probability and we can select an action with 1-epsilon probability that gives maximum reward in given state. For example, if an experiment is about to run 10 times. This policy selects random actions in twice if the value of epsilon is 0.2.
Consider a following example, There is a robot with capability to move in 4 direction. Up,down,left,right. You can write a policy which selects a direction which gives maximum reward with 1-epsilon probability and with epsilon probability it select random direction. Here epsilon allows robot to explore new path.
Now the question is how to implement it.
Formula of estimated probability of an event is

probability of an event= number of times event occurs/(total number of trails) ;

We know the epsilon (the probability of an event should occur) . And if we know the number of trails N then we can find number of times events should happen during N trails. so the pseudo code for implementing e-greedy policy can be written as

```
counter=0;
epsilon=0.2;
maxtrails=epsilon*nTrails;
for(int i=0;i<nTrails;i++)
{
if(counter<maxtrails)
selectrandom();
counter++;
else
selectmaxaction();
}
```

More sophisticated code can be written that selects random action for given probability with different permutation.But there is a one more elegant way to implement e-greedy policy.
Look at the code below and think does it make sense?

```
for (int i=0;i<nTrails;i++)
{
// function rand() returns random-number between 0 to 1 with uniform distribution
float temp=rand() ;
if(temp<epsilon)
selectrandom();
else
selectmaxaction();
}
```

The trick is to use the classical probability definition. Definition says that

true probability of an event = (number of favorable outcomes of event)/ (number of possible outcomes) .
if number of trails approaches to infinity.

so the next question could be what is a probability that selected random-number is less than epsilon.
The answer is

total numbers which are less that epsilon/ (total numbers).

If a function rand() returns random number between 0 and 1 with interval of 0.1. for example 0,0.1,0.2,0.3,…0.8,0.9. and the value of epsilon is 0.2 then the estimated probability for selected number is less than 0.2 is (2/10)= 0.2 as per definition.
And probability of selecting random action is equal to the probability of selecting random number less than epsilon as per above code. So theoretically the code above selects the random action with epsilon probability.

We can verify the above argument by following code.

```
void greedy()
{
ep=0.2;
rn=0;
mx=0;
nTrails=1000;
for (int i=0;i<nTrails,i++)
{
m=rand();
if(m<ep)
rn=rn+1;
else
mx=mx+1;
}
printf('probability of random action selected given no.trails %d is %f\n',nTrails,rn/nTrails);
}
```

Below is the output when function greedy calls for five times with nTrails equals to 1000 and 100000 . As it can be seen that number of trails increasing estimated probability is approaches to true probability.

```
probability of random action selected given no.trails 1000 is 0.214000
probability of random action selected given no.trails 1000 is 0.189000
probability of random action selected given no.trails 1000 is 0.164000
probability of random action selected given no.trails 1000 is 0.211000
probability of random action selected given no.trails 1000 is 0.216000

probability of random action selected given no.trails 100000 is 0.199320
probability of random action selected given no.trails 100000 is 0.198820
probability of random action selected given no.trails 100000 is 0.199450
probability of random action selected given no.trails 100000 is 0.198390
probability of random action selected given no.trails 100000 is 0.197960
```

USA for UNHCR The UN Refugee Agency
3 years old — and in grave danger.
Help Syria's children now.
MAKE YOUR EMERGENCY GIFT

© UNHCR/Andrew McConnell
USA for UNHCR The UN Refugee Agency
What happens to her, when violence destroys her home?
YOU decide.
MAKE YOUR EMERGENCY GIFT

Tags: classic probability, e-greedy policy, epsilon policy, estimated probability, Markov process, q learning, reinforcement learning, relative probability, simulator

Comments RSS (Really Simple Syndication) feed

1 Comment:

- Bhanu Chander
  March 8, 2015 at 4:34 pm
  Thank you for the explanations

  Reply

3 Trackbacks / Pingbacks for this entry:

- When to Run Bandit Tests Instead of A/B/n Tests
  […] epsilon-greedy is a constant play […]
- When to Run Bandit Tests Instead of A/B/n Tests - InfoPult News
  […] epsilon-greedy is a constant play […]
- When to Run Multi Armed Bandit Tests Instead of A/B/n Tests | James Caldwell

[…] epsilon-greedy is a constant play […]