MAS 19-20 HW 1 Solutions

Multi-agent systems (Vrije Universiteit Amsterdam)

Multi-Agent Systems

Homework Assignment 1

MSc AI, VU

E.J. Pauwels

Version: Nov 1, 2019 — Deadline: Nov 7, 2019 (23h59)

1 Monte Carlo simulation

1.1 MC sampling

Recall that Monte Carlo sampling allows us to estimate the expectation of a random function by sampling from the corresponding probability distribution. More precisely, if $f(x)$ is a 1-dim (continuous) probability density, and $X \sim f$ is a stochastic variable distributed according to this density f, then the expected value of some function φ can be estimated using Monte Carlo sampling by:

$$E_f(\varphi(X)) \equiv \int \varphi(x)f(x)\,dx \approx \frac{1}{n}\sum_{i=1}^{n}\varphi(X_i) \qquad \text{for sample of independent } X_1, X_2, \ldots, X_n \sim f.$$

1. Assume that $X \sim N(0,1)$ is standard normal. Estimate the mean value $E(\cos^2(X))$. Quantify the uncertainty on your result.

2. Suppose you're designing a deep neural network that needs to maximize some score function S. The actual design of the network depends on some hyperparameter A. Training the networks is computationally very demanding and time consuming, and as a consequence you have only been able to perform ten experiments to date. Based on these ten data points you observe a slight positive correlation of 0.3 between the value of the hyperparameter A and the score S. If this result is genuine, it suggest to increase A in the next experiment in order to improve the score. But if the correlation is not significant, increasing A could lead you astray. How would you use MC to decide whether the correlation is significant?
Hint: Compute the empirical p-value of the observed result, under the assumption of independence.

1.2 Importance Sampling

Importance sampling extends the basic MC approach to cases where it is difficult to sample from f but (relatively) easy to sample from a (somewhat) similar distribution g. More precisely:

$$
\begin{aligned}
E_f(\varphi(X)) &= \int \varphi(x) f(x)\, dx \\
&= \int \varphi(x) \frac{f(x)}{g(x)}\, g(x)\, dx \equiv E_g\left[\varphi(X)\frac{f(X)}{g(X)}\right] \\
&\approx \frac{1}{n}\sum_{i=1}^{n} \varphi(X_i)\frac{f(X_i)}{g_(X_i)} \qquad \text{for sample of independent } X_1, \ldots, X_n \sim g.
\end{aligned}
$$

1. Let $X \sim N(0,1)$ be a standard normal stochastic variable. Use importance sampling to estimate $E(X^2)$ by sampling from a uniform distribution $q \sim U(-5,5)$ on the interval $[-5,5]$. What value do you expect (based on your knowledge of the normal distribution)? How accurate is your estimate based on importance sampling?

2. Suppose some random process produces output ($-1 \leq X \leq 1$) that is distributed according to the following continuous density:

$$
f(x) = \frac{1+\cos(\pi x)}{2} \qquad (\text{for } -1 \leq x \leq 1).
$$

Again we are interested in estimation $E(X^2)$. However, as this is not a standard distribution it makes sense to use importance sampling to estimate this value. Quantify the uncertainty on your result.

1.3 Kullback-Leibler divergence

The Kullback-Leibler (KL) divergence measures quantifies the similarity (or dissimilarity) of two probability densities. More specifically, given two continuous (1-dim) probability densities f, g, the KL-divergence is defined as:

$$
KL(f||g) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)}\, dx \quad \equiv \quad E_f\left[\log\left(\frac{f(X)}{g(X)}\right)\right] \tag{1}
$$

1. Let $f \sim N(\mu, \sigma^2)$ and $g \sim N(\nu, \tau^2)$ both be normal distributions. Express $KL(f||g)$ as a function of the means and variances of f and g. We mention in passing that the KL expression in eq.1 is called a **divergence** rather than a **distance** because it's not symmetric. Use the expression obtained above to convince yourself of this fact.

2. Check your theoretical result in (1) by computing a sample-based estimate of the KL-divergence (Monte Carlo simulation). Pick an appropriate sample size. Compare the MC estimate to the theoretical result.

2 Exploitation versus Exploration

2.1 UCB versus ϵ-greedy

Write a programme to experiment with the exploration/exploitation for the k-bandit problem (e.g. take $5 \leq k \leq 20$). Assume that the arms generate normally distributed rewards. Produce graphs to compare the average reward (over time) for different strategies (ϵ-greedy, greedy with optimistic initialisation, UCB) *No need to submit code, only the results.*

2.2 The intuition behind Lai-Robbins lower bound for expected total regret

NOTE: for this problem, we are not looking for a formal proof, an intuitive argument suffices.

In the k-bandit problem we assumed that each arm a generated a random reward according to the a density distribution f_a with fixed but unknown mean $q(a)$. If $q^* = \max_a q(a)$ is the maximal average reward, then $\Delta_a := q^* - q(a)$ is the opportunity gap for arm a, i.e. the amount you lose by playing arm a rather than the optimal arm. Recall that

- $N_t(a)$: number of times action a has been selected up till time t;

$$N_t(a) = \sum_{i=1}^{t} 1_{(A_i=a)} \quad \implies \quad EN_t(a) = \sum_{i=1}^{t} P(A_i = a)$$

- Expected total regret (at time t):

$$L_t = E\left(\sum_{i=1}^{t}(q^* - q(A_i))\right) = \sum_a \Delta_a \left\{\sum_{i=1}^{t} P(A_i = a)\right\} = \sum_a \Delta_a \, EN_t(a)$$

Now consider the simplest possible case: a 2-armed bandit such that $q_1 > q_2$, i.e. $\Delta_1 = 0$ and $\Delta_2 = q_1 - q_2 > 0$. Suppose you play the game for an unlimited (infinite) number of times, so we are interested in the behaviour of L_t for large values of t.

1. Why is it not optimal to stop sampling one of the arms after a certain time? Put differently, each arm should be sampled an infinite number of times (as $t \to \infty$). Why?

2. Further to the above, we want to ensure that for the second arm ($a = 2$) the expected number of plays $EN_t(a) \to \infty$ as $t \to \infty$. Compare the following sampling schemes for arm $a = 2$:

$$(i)\, P(A_t = 2) = \frac{1}{t} \qquad (ii)\, P(A_t = 2) = \frac{1}{t^2} \qquad (iii)\, P(A_t = 2) = \frac{1}{t^{1+\epsilon}} \quad (\epsilon > 0).$$

   Can you see why (for large values of t) the expected total regret L_t would have a lower bound that is proportional to $\log t$.

SOLUTIONS

1.1 Mont Carlo Sampling

Estimate $E(\cos^2(X))$ for $X \sim N(0,1)$: Matlab code

```
sample_size =  10000;    %  sample size for MC estimate

X = randn(sample_size,1);  %  random sample from N(0,1) population
F = cos(X).^2;     %  compute function value at each sample point

%  The MC estimate for m = E((cos(X))^2) is obtained by computing the
%  sample average:

m_mc = mean(F);

%  Since each sample point in F is and independent sample from cos(X).^2
%  (where X ~ N(0,1), the standard deviation  std(F) is an estimate of the
%  corresponding population standard deviation.  The corresponding standard
%  deviation for the sample mean is therefore equal to std(F)/sqrt(sample_size)

m_mc_std = std(F)/sqrt(sample_size);
```

Correlation between score and hyperparameter

- Assume that there is no correlation;
- Use this assumption to draw random samples (of size 10) from this distribution and compute the correlation coefficient.
- Compare the observed result $r_{obs} = 0.3$ to the correlations for the simulated samples. Compute how "extreme" the observed result is (compute its p-value). If the p-value is small (e.g. $p < 0.05$) the observed trend is likely to be genuine.

MATLAB code:

```
%  We have 10 data points for which the observed correlation equals r_obs = 0.3.

r_obs = 0.3;
n = 10;   % number of experimental data points

%  Assume that there is no correlation between the two parameters, then the
%  observed correlation is a random fluctuation.  To test how likely this
%  size of fluctation is, we generate independent variables and tally how
%  often a correlation of r_obs (or larger) is observed.

nr_samples = 1000;
```

```
Rho_MC = zeros(nr_samples,1);

for i = 1:nr_samples
    %  Generated randomly distributed but independent samples for S and A
    S = randn(n,1);
    A = randn(n,1);
    % Compute and store the observed correlation coef for each sample
    Rho  = corrcoef(A,S);   % full correlaton matrix
    Rho_mc(i) = Rho(1,2) ;   % correlation is off-diagonal element of corr matrix

end

%  Compute the p-value of the observed value

pval = length(find(Rho_mc > r_obs))/nr_samples;
```

1.2 Importance Sampling

Matlab code for estimating EX^2 using samples from uniform

```
%  X ~ N(0,1), hence density  = f(x) = 1/sqrt{2\pi}  exp(-x^2/2)
%     Density for uniform U(-5,5) :  g(x) = 1/10;
%
%  We need to estimate  EX^2 = Var(X) = 1 by sampling from the uniform;
%
%  This means that we need to sample say U ~ U(-5,5) and compute the sample
%  value :
%      F = phi(U) (f(U)/g(U))   where phi(u) = u^2

sample_size = 1000

U = 10*rand(sample_size,1)-5;

F = (U.^2) .* (10*normpdf(U));

mc_estimate = mean(F);
mc_population_std = std(F);
mc_estimate_std = std(F)/sqrt(sample_size)
```

Matlab code for estimating from unusual distribution

```
%  Question 2:
%--------------
```

```
%  X is distributed according to density   f(x) = (1+cos(pi*x))/2;

dx = 0.01;
xx = (-1:dx:1);

f = @(x) (1+cos(pi*x))/2;  % density

sample_size = 1000   % for MC sample

%  Sample from uniform

U = 2*rand(sample_size,1)-1;    %  Uniform on -1, 1;  density = 1/2

F = (U.^2) .* (2*f(U));  % compute the pointwise result;

mc_estimate = mean(F);
mc_population_std = std(F);
mc_estimate_std = std(F)/sqrt(sample_size)
```

1.3 Kullback-Leibler divergence

KL for two gaussians Assuming normal densities $f \sim N(\mu_1, \sigma_1^2$ and $g \sim N(\mu_2, \sigma_2^2$, a straightforward computation yields:

$$KL(f||g) = \int f(x) \log \frac{f(x)}{g(x)}\, dx = \log \frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}.$$

Notice the asymmetric role of both densities. Although not obvious from the above, the KL distribution is always non-negative.

MC for KL estimation

```
% f ~ N(mu1,sigma1^2)
mu1 = 0;    sigma1 = 2;

% g ~ N(mu2,sigma2^2)
mu2 = 2;    sigma2 = 3;

sample_size = 1000

%  Sample from f and compute the KL value at each sample point

X = mu1 + sigma1*randn(sample_size,1);
KL = log(normpdf(X,mu1,sigma1)./normpdf(X,mu2,sigma2));
```

```
KL_div = mean(KL);
KL_div_std = std(KL)/sqrt(sample_size);

% KL divergence based on theoretical expression:

KL_div_theory = log(sigma2/sigma1) + (sigma1^2+(mu1-mu2)^2)/(2*sigma2^2) - 1/2;
```

2.2 Lai-Robbins intuition

1. If one stops exploring after a finite time, there is a small but non-zeros risk of settling on the sub-optimal arm, after which one accrues a constant amount of regret at every step. Put differently, one needs to choose a sampling regime that ensures that number of visits for each arm is not bounded.

2. Sampling scheme (ii) is a special case of (iii), and we can easily check that for any $\epsilon > 0$

   $$\lim_{T\to\infty} EN_T(a=2) = \lim_{T\to\infty} \sum_{t=1}^{T} P(A_t = 2) = \lim_{T\to\infty} \sum_{t=1}^{T} \frac{1}{t^{1+\epsilon}} < \infty.$$

   However if we let $\epsilon \to 0$ we end up with sampling scheme (i) for which:

   $$EN_T(a=2) = \sum_{i=1}^{T} \frac{1}{t} \approx \log T \quad \text{as } T \to \infty$$

   So it turns out that in this case the total expected regret depends logarithmically on the number of steps t:

   $$L_t \sim \log t \qquad \text{as } t \to \infty.$$