Qishi Quiz 3

Yongjie Xin

June 23, 2019

Abstract

Results and answers.

Contents

1 Math

1.1 Q1

The event I_A is the number of girl neighbors for a random boy A. The following is the possible numbers of I_A and also its probability;

$$I_A = \begin{cases} 0, & \binom{8}{2} \Big/ \binom{15}{2} \\ 1, & \binom{8}{1} * \binom{7}{1} \Big/ \binom{15}{2} \\ 2, & \binom{7}{2} \Big/ \binom{15}{2} \end{cases} \tag{1}$$

So the expectation $E[I_A] = 14/15$. Since A is a random boy, for all the boys the number of girl neighbors are $42/5$. Any girl neighbor for a boy is also a boy neighbor for a girl.

1.2 Q2

Independent is definite uncorrelated. Uncorrelated is not enough to be independent. For example, x is normal; The correlation between x and x^3 is zero. But apparently they are not independent.

1.3 Q3

American tree, optimal early exercise; For one single toss, the expectation is 3.5. So if we are doing the second toss, and only one toss left; If we get 4, 5, $6 > 3.5$, we stop; if we get 1, 2, 3, we continue and expect to get 3.5; The expected gain for a second toss is $1/6 * (4 + 5 + 6) + 3.5 * 1/2 = 4.25$. For the first toss, if we could get 5, $6 > 4.25$, we stop; otherwise, we continue, we expect to get 4.25; So $1/6 * (5 + 6) + 4/6 * 4.25 = 14/3$.

1.4 Q4

Solution quoted from mathStack The inclusion/exclusion principle explains quite well that the Union of all the events could be broken into categories. Here I just wanna add one additional point for understanding $P(V_i > x) = (1-x)^{n-1}$. If we think of the rod as a loop, then it is more natural to see, we could view the left hand side of V_i as the zero point; So $P(V_i > x)$ is just probability for all the other $n-1$ points falls within $[x, 1]$. Same explanation could be used for $P(V_i > x \, and \, V_j > x)$.

1.5 Q5

The correlation matrix has to be positive semi-definite; (The variance of a weighted sum $\sum w_i X_i$ of random variables must be non-negative; This is the same as the definition of positive semi-definite matrix. Without losing generality, X_i are standardized.)

$$\begin{pmatrix} 1, & r, & 0 \\ r, & 1, & r \\ 0, & r, & 1 \end{pmatrix} \tag{2}$$

so that $1 - 2r^2 \geq 0 \Rightarrow |r| < \sqrt{2}/2$.

1.6 Q6

Use X_t to denote the man's location after t moves. X_t is a martingale, since $E[X_{t+1}|X_t] = 0.5(X_t + 1) + 0.5(X_t - 1)$. Denote τ as the stopping time, which is either exist from left door or right door. Martingale optimal stopping theorem: $E[X_\tau] = E[X_0]$.

$$P_L * 0 + (1 - P_L) * 100 = 1 \tag{3}$$

$$P_L = 99/100 \tag{4}$$

P_L is the probability to leave from left door.

To get the expected number to leave is to calculate $E[\tau]$. $X_t = X_0 + Z_1 + Z_2 + Z_3 + ... + Z_t$;

$$\sum_1^t Z_i = X_t - X_0 \tag{5}$$

$$E[Z_i] = 0, \quad E[Z_i^2] = 1, \quad E[Z_i Z_j] = 0 \tag{6}$$

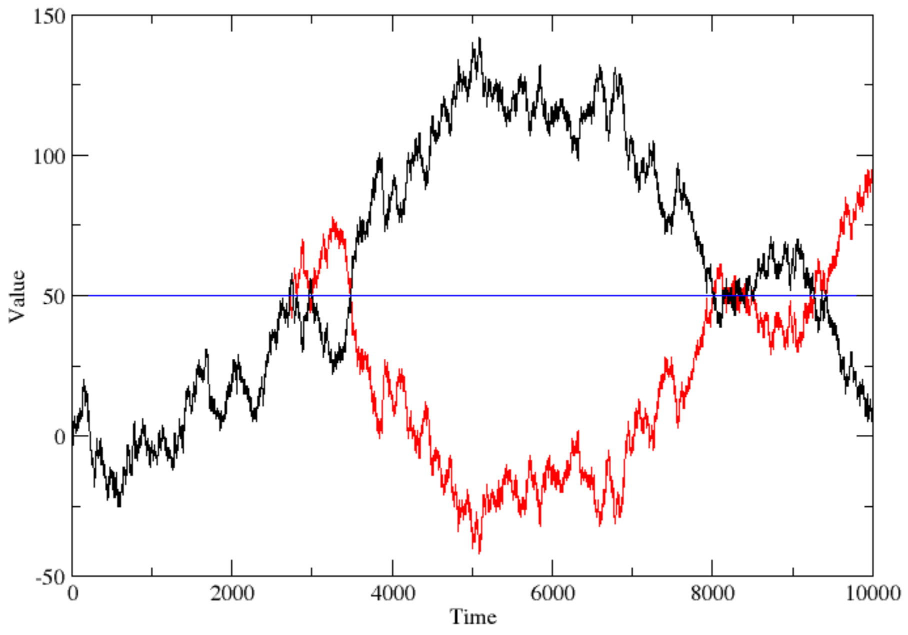

Figure 1: Reflection principle of Brownian motion; Source Wikipedia

Wald's second equation

$$E[(\sum_{1}^{\tau} Z_i - \tau E[Z_1])^2] = E[\tau]Var[Z_1] \tag{7}$$

So that

$$E[\tau] = 99^2 * 1/100 + 1^2 * 99/100 = 99 \tag{8}$$

part b Brownian motion reflection principle, any path reaches -100, having the same probability reaching +100. The game rule is now left door is at -100, what are the expected number of moves? Still use the above method:

$$E[\tau] = 99^2 * 101/200 + 101^2 * 99/200 = 99 * 101 \tag{9}$$

1.7 Q7, and Q8

Both Ridge Regression and Lasso are techniques for analyzing regression data that suffer from multicollinearity. When there is multicollinearity, least square estimates are unbiased, but variances are large. By adding a degree of bias, the two could reduce the standard errors, hoping the net effect will be more reliable.

Both are shrinkage methods:

- Ridge, uses quadratic shrinking
- Lass, uses absolute-value shrinking.

Ridge Regression

$$min_\beta (y - X\beta)^T (y - X\beta) \quad subject\,to \quad \sum \beta_j^2 \leq s \tag{10}$$

In Lagrange form:

$$min_\beta \big\{ (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \big\} \tag{11}$$

Lasso Regression in Lagrange form:

$$min_\beta \big\{ (y - X\beta)^T (y - X\beta) + \lambda |\beta| \big\} \tag{12}$$

1.8 Q9

IMHO, I don't see anything wrong with the game. about after 20 throws, if ever, I get a head, I will be paid more than a million.

Give it another thought, if the rule is if I toss a head, the game stops. Then that's another game. The expectation is infinity, since if the game stops after ith toss, the expected payoff is $(1/2)^i * 2^i = 1$. Sum all the 1, the total expectation is infinity.

If we could play the game infinite times, we could reach break-even; Just 1M is too much. Given the total money printed in the world, we can't play the game infinite times. Another consideration is a coin toss takes time, one might have to play until death, but still in debt.

2 Programming

2.1 Q10

Default for c++ is a key word to ask the compiler to generate default constructor, copy constructor, destructor.

- Whenever we declare a parameterized constructor, the compiler won't create a default constructor. This case we have to explicitly write the default constructor.

My reference GeekForGeeks

2.2 Q11

Singleton, is design pattern everyone loves to hate. Its advantange is obvious; if we need to load a Data base, we just want to load it once; who ever uses it, access it from this copy only. We might want to disable copy or even modification.

Official definition is Singleton is a design pattern that restricts the instantiation of a class to one object.

in C++ 11, Meyers's Singleton is Thread safe. Lazy initialization

```
static Singleton& instance()
{
     static Singleton s;
     return s;
}
```

Quoted from the standard:

```
If control enters the declaration concurrently
 while the variable is being initialized,
 the concurrent execution shall wait for
 completion of the initialization.
```

This is only available after and including VS2015. Another approach is to use standard function

```
std::call_once
```

This ensures a callable is only exected exactly once.

```
// singletonCallOnce.cpp

```

```
#include <chrono>
#include <iostream>
#include <future>
#include <mutex>
#include <thread>

constexpr auto tenMill= 10000000;

class MySingleton{
public:
  static MySingleton& getInstance(){
    std::call_once(initInstanceFlag, &MySingleton::initSingleton);
    // volatile int dummy{};
    return *instance;
  }
private:
  MySingleton()= default;
  ~MySingleton()= default;
  MySingleton(const MySingleton&)= delete;
  MySingleton& operator=(const MySingleton&)= delete;

  static MySingleton* instance;
  static std::once_flag initInstanceFlag;

  static void initSingleton(){
    instance= new MySingleton;
  }
};

MySingleton* MySingleton::instance= nullptr;
std::once_flag MySingleton::initInstanceFlag;
```

My reference Singleton

2.3 Q12

My answer in Quiz2 of Q17 had a typo. Dynamic programming:

```
f(k, i) is the max payoff, after k transactions at stock i;
f(k, i) = max{ f(k, i-1) ,
( [f(k-1, j) + Stock(i) - Stock(j)], j from 0 till i ) };
```

First initialize a 2d array/matrix. Row index is the number of transaction. index equals zero, is the first transaction. Column index is the index of the stock quote, or time index, flowing from left till right.

2.4 Q13

My initial thought was using a hash map, with integer value as hash, count as value. it takes O(n) memory. Another post online shows using XOR operation.

```
(1) XOR of a number with itself is 0.
(2) XOR of a number with 0 is itself.
```

```
using namespace std;

int findSingle(int ar[], int ar_size)
    {
        // Do XOR of all elements and return
        int res = ar[0];
        for (int i = 1; i < ar_size; i++)
            res = res ^ ar[i];

        return res;
    }
```

My reference IntegersTwiceOnce.

2.5 Q14

It is not ok. Based on the polymorphism of c++; for a base pointer, pointing to derive class, at running time, the non-virtual function of base is called, inside the non-virtual base function, the virtual function of derived class is executed (instead of the base version). My reference virtualNonVirtual.

2.6 Q15

True random numbers are generated from radioactive decay, based on quantum mechanics. For computers, it could be done using Linear congruential generator.

Quality

- how long is the sequence. Pseudo random numbers will repeat itself after enough long time.

- The true randomness. It is hard to find sufficient tests. Some of the necessary statistical tests could be symmetrical around median. Chi-square test...

2.7 Q16

Simulate a dice with fair coin. Consecutive 3 flips, and assign the 8 kinds of outcomes to 1, 2, 3, 4, 5, 6. and Reject two of them. the division of $2^n/6$ is not exact. So in my opinion, rejection is unavoidable. Here we reject 2/8 cases, 25% rejection. Of course, one could flip more than 3 times, maybe n times, then do assignment and rejection. The rejection rate could be lower, however there is a cost for coin flip. If coin flip is worthless, a larger n could be better. So we are finding a n flip that

$$min\Big\{FlipCost(n) + RejectionCost(2^n/6)\Big\} \tag{13}$$

Rejection cost is a flip cost, which to re-flip n times again.

2.8 Q17

My reference twoRobots. The logic is to use the zero axis as a reference point, whoever reaches it first will speed up. This guarantees the meetup.