2011 Winter BIOSTAT 615/815 Final Exam

Thursday April 21st, 10:30AM-12:30PM

NAME : ( )

Problem 1. True/False Questions (20 pts - 2pts each)

Write down **T** (True) or **F** (False) for each of the statement below.

(a) ( ) RadixSort is always faster than comparison-based sort algorithms.

(b) ( ) With sufficiently large amount of memory, the expected time complexity of search from a hash table is constant.

(c) ( ) Golden search has a better worst-case time performance than optimization using parabolic interpolation.

(d) ( ) Simplex method does not require to have derivate of the function to optimize

(e) ( ) Simplex algorithm always converges to the optimal point when the function is convex

(f) ( ) E-M algorithm always guarantees to find global optima

(g) ( ) Both Simulated Annealing and Gibbs Sampling are instances of the Metropolis-Hastings algorithm.

(h) ( ) Gibbs sampling allows us to empirically estimate the joint posterior distribution of the parameters given observed data.

(i) ( ) Importance sampling can reduce the variance of the estimation compared to crude Monte-Carlo methods.

(j) ( ) Baum-Welch algorithm is a polynomial-time algorithm.

Problem 2. Short answer questions (10pts - 5pts each)

(a) Suppose that we have a uniformly sampled true random number $x \in [0, 1]$. What additional function is needed to generate a random sample from a probability distribution? Also, describe how the random sample can be generated.

(b) Consider the following 2-state Markov Chain with the following transition matrix. What is the stationary distribution of the states?

$$\begin{pmatrix} q_1^{(t+1)} \\ q_2^{(t+1)} \end{pmatrix} = \begin{pmatrix} 0.95 & 0.2 \\ 0.05 & 0.8 \end{pmatrix} \begin{pmatrix} q_1^{(t)} \\ q_2^{(t)} \end{pmatrix}$$

Problem 3. Dynamic Polymorphism (10 pts)

Write down the expected output for the following program, next to the corresponding line in the main function

```
#include <iostream>
class rect {
public:
  int x, y;
  rect(int _x = 0, int _y = 0) { x = _x; y = _y; }
  int area() { return x*y; }
  std::string whoAmI() { return std::string("rect"); }
  virtual std::string virtualWhoAmI() { return std::string("rect"); }
};

class square : public rect {
public:
  square(int _x = 0) { x = _x; y = _x; }
  std::string whoAmI() { return std::string("square"); }
  virtual std::string virtualWhoAmI() { return std::string("square"); }
};

void printRectByRef(rect& r) {
  std::cout << r.whoAmI() << "\t" << r.virtualWhoAmI() << "\t" << r.area() << std::endl;
}

void printRectByVal(rect r) {
  std::cout << r.whoAmI() << "\t" << r.virtualWhoAmI() << "\t" << r.area() << std::endl;
}

int main() {
  rect x1(1,2);
  square x2(3);
  rect x3 = x2;
  rect* px4 = new square(4);
  square* px5 = new square(5);

  printRectByRef(x1);
  printRectByRef(x2);
  printRectByRef(x3);
  printRectByRef(*px4);
  printRectByRef(*px5);

  printRectByVal(x1);
  printRectByVal(x2);
  printRectByVal(x3);
  printRectByVal(*px4);
  printRectByVal(*px5);

  return 0;
}
```

Problem 4 - Generating a Random Permutation (10pts)

Consider the following function `myShuffle()`

```
void myShuffle(double* array, int size) {
  double tmp;
  for(int i = size-1; i > 0; --i) {
    j = trueIntRandom(0,i);
    tmp = a[i];
    a[i] = a[j];
    a[j] = tmp;
  }
}
```

Assuming the function '`int trueIntRandom(int min, int max)`' provides a true random number between `min` and `min` (including the boundary values), does `myShuffle()` function produces a random permutation of the input array `array` of size `size`? If yes, prove it mathematically. If no, provide counter examples.

Problem 5. Monte Carlo Methods (20pts)

Consider the following algorithm NaiveWeightedSampler

Data: $\mathbf{w} = (w_1, \cdots, w_k)$ where $\sum_{i=1}^{k} w_i = 1$
while ***true*** **do**
  Generate a random integer $i \in \{1, \cdots, k\}$ from a uniform distribution.
  Generate a random real number $p \in [0, 1]$ from a uniform distribution.
  If $p < w_i$, then return i and exit the loop, otherwise repeat the loop.
end

(a) (5pts) Prove that the algorithm NaiveWeightedSampler produces a random integer $x \in \{1, 2, \cdots, k\}$ following a distribution $\Pr(x = i) = w_i$ for $i \in \{1, \cdots, k\}$.

(b) (5pts) How many draws of random numbers are expected to be made to sample one random number using NaiveWeightedSampler?

(c) (10pts) Write an algorithm SmartWeightedSampler that requires only a single random draw of a real number $p \in [0, 1]$ from a uniform distribution (without the need of any additional random number) in order to produce a random integer $x \in \{1, 2, \cdots, k\}$ following the distribution $\Pr(x = i) = w_i$ for $i \in \{1, \cdots, k\}$.

Problem 6. Continuous Time Markov Process (20pts)

Consider a symmetric two-state continuous-time markov process, where the probability of no-transition between an arbitrary time interval $(t, t+r)$ is defined as

$$\Pr(q_s = i, \quad \forall s \in (t, t+r) | q_t = i) = \exp(-\lambda r)$$

where $i \in \{1, 2\}$.

(a) (5pts) If $\Pr(q_t = 1) = \Pr(q_t = 2) = 1/2$ holds for arbitrary time $t \geq 0$, what is $\Pr(q_{t+r} = i | q_t = i)$?

(b) (5pts) What is the expected number of state transitions between t and $t + r$?

(c) (10pts) Suppose that you have observed a list of states $\mathbf{q} = (q_0, q_1, \cdots, q_n)$ at time points $\mathbf{t} = (0, t_1, t_2, \cdots, t_n)$. Write down $f(\lambda; \mathbf{t}, \mathbf{q}) = \log \Pr(\mathbf{q} | \lambda, \mathbf{t})$. Suppose that we want to find $\arg\max_\lambda f(\lambda; \mathbf{t}, \mathbf{q})$. Among the optimization methods described in the class, which optimization method would be useful to obtain the MLE parameter λ? Describe how to estimate it numerically. You don't need to provide actual code, but please describe the procedure (or psuedo-code) in detail enough to be unequivocally implemented.

Problem 7. E-M algorithm, Simulated Annealing, and Gibbs Sampler (10pts)

Describe key advantages and disadvantages of E-M algorithm, Simulated Annealing, and Gibbs Sampler to the best of your knowledge. You may provide exemplified description based on your experience with your homework or problems discussed in the class if you would like to.