Jinyang Ruan

011696096

CPT_S 515 Advanced Algorithms HW#4

1. Step 1: For each walk $\alpha$ from initial to final. We use $\#_p(\alpha)$ to denote the number of pump instruction $p$ on the $\alpha$.

   Step 2: For each $\#_p(\alpha)$ on the $\alpha$, we can obtain a formula:

$$W_p(x_1, x_2, \dots, x_k, \#_p(\alpha))$$

   Step 3: After running the entire $\alpha$, the water levels (from 0's) satisfy:

$$\exists\ x_{1,p}, \dots, x_{k,p}\ , p \in pump\ instructions\ on\ M$$

$$x_1 = \sum_p x_{1,p} \wedge$$

$$x_2 = \sum_p x_{2,p} \wedge$$

...

$$x_k = \sum_p x_{k,p} \wedge$$

$$\forall p\text{: } W_p(x_{1,p}, x_{2,p}, \dots, x_{k,p}, \#_p(\alpha))$$

   Step 4: Denote the formula above as:

$$W_\alpha(x_1, x_2, \dots, x_k)$$

   For each fixed $\alpha$, $W_\alpha$ is a Boolean combination of linear constraints over real variables.

   Step 5: Let $\Gamma$ be the set of all $\alpha$ from initial to final. For each $\alpha$, we have a formula for water levels:

$$W_\alpha(x_1, x_2, \dots, x_k)$$

Step 6: I use vector $\#(\alpha)$ to denote the number of different pump instructions on the $\alpha$.

For all $\alpha \in \Gamma$, we obtain a set of vectors $V = \{\#(\alpha): \alpha \in \Gamma\}$

Step 7: There is an integer linear constraints system $Q(v)$, s.t.

$$\forall v, v \in V \quad iff\ Q(v)\ holds$$

Now, for each $\alpha \in \Gamma$, we have $W_\alpha(x_1, x_2, \dots, x_k)$.

For all $\alpha$, we have $V = \{\#(\alpha): \alpha \in \Gamma\}$, that is characterized by $Q$.

Step 8: Combine $Q$ and $W_\alpha(x_1, x_2, \dots, x_k)$, we get an integer linear constraints system:

$$W_\alpha\left(x_1, x_2, \dots, x_k, \#(\alpha)\right).$$

Step 9: If $\exists \alpha, s.t. W_\alpha\left(x_1, x_2, \dots, x_k, \#(\alpha)\right)$ and $Bad(x_1, x_2, \dots, x_k)$ are satisfied, we can say $M$ has a bad walk.

2. Step 1: If it is a fair dice, we expect the frequency of each number to be 1 / 6.

   Step 2: Build a table to record all results of tossing as shown below:

| Outcomes (number) | Occurrences ($A_i$) | Expectations ($E_i$) |
|---|---|---|
| 1 | $x_1$ | 1000 / 6 |
| 2 | $x_2$ | 1000 / 6 |
| 3 | $x_3$ | 1000 / 6 |
| 4 | $x_4$ | 1000 / 6 |
| 5 | $x_5$ | 1000 / 6 |
| 6 | $x_6$ | 1000 / 6 |

   Step 3: Use Chi-Square test

$$\chi^2 = \sum_{i=1}^{6} \frac{(A_i - E_i)^2}{E_i}$$

Step 4: Search the table of Chi-Square distribution. We look up the value of

$$df = k - 1 = 6 - 1 = 5$$

| Degrees of freedom (df) | $\chi^2$ value[19] | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.004 | 0.02 | 0.06 | 0.15 | 0.46 | 1.07 | 1.64 | 2.71 | 3.84 | 6.63 | 10.83 |
| 2 | 0.10 | 0.21 | 0.45 | 0.71 | 1.39 | 2.41 | 3.22 | 4.61 | 5.99 | 9.21 | 13.82 |
| 3 | 0.35 | 0.58 | 1.01 | 1.42 | 2.37 | 3.66 | 4.64 | 6.25 | 7.81 | 11.34 | 16.27 |
| 4 | 0.71 | 1.06 | 1.65 | 2.20 | 3.36 | 4.88 | 5.99 | 7.78 | 9.49 | 13.28 | 18.47 |
| 5 | 1.14 | 1.61 | 2.34 | 3.00 | 4.35 | 6.06 | 7.29 | 9.24 | 11.07 | 15.09 | 20.52 |
| 6 | 1.63 | 2.20 | 3.07 | 3.83 | 5.35 | 7.23 | 8.56 | 10.64 | 12.59 | 16.81 | 22.46 |
| 7 | 2.17 | 2.83 | 3.82 | 4.67 | 6.35 | 8.38 | 9.80 | 12.02 | 14.07 | 18.48 | 24.32 |
| 8 | 2.73 | 3.49 | 4.59 | 5.53 | 7.34 | 9.52 | 11.03 | 13.36 | 15.51 | 20.09 | 26.12 |
| 9 | 3.32 | 4.17 | 5.38 | 6.39 | 8.34 | 10.66 | 12.24 | 14.68 | 16.92 | 21.67 | 27.88 |
| 10 | 3.94 | 4.87 | 6.18 | 7.27 | 9.34 | 11.78 | 13.44 | 15.99 | 18.31 | 23.21 | 29.59 |
| **P value (Probability)** | 0.95 | 0.90 | 0.80 | 0.70 | 0.50 | 0.30 | 0.20 | 0.10 | 0.05 | 0.01 | 0.001 |

Step 5: Usually we choose P value is 0.05, i.e. 95% confidence level.

In this case, if $\chi^2 \leq 11.07$, we consider the dice is fair. If $\chi^2 > 11.07$, the dice is unfair, and when the value of $\chi^2$ is bigger, we consider the dice is more “unfair”.

3. Step 1: Encode each symbol from alphabet as a unique non-binary number. Assume there are $m$ sequences in set $S$. Combine sequences in pairs, then we have $\frac{m^2}{2}$ pairs. Denote a variable $P = 0$ to store the number of i.i.d. pairs.

   Step 2: We have a new set $S'$, all sequences in set $S'$ now are represented as number sequences. Calculate the average values for each sequence:

$$\bar{x}_1 = \frac{\sum_{i=1}^{n} x_{1,n}}{n}$$

...

$$\bar{x}_m = \frac{\sum_{i=1}^{n} x_{m,n}}{n}$$

Pick one pair in $S'$. Hypothesize those two sequences are identically distributed. Apply permutation test. Choose the threshold value $p = 0.05$. If the hypothesis is not true, repeat step 2, choose other sequences pairs. If the hypothesis in step 2 is true, hypothesize they are independent.

Step 3: Apply Chi-square test. If p-value is less than 0.05, hypothesis is not true, repeat step 2, choose other sequences pairs. If p-value is larger than 0.05, hypothesis is true. $P = P + 1$.

Step 4: After going through all pairs in $S'$, the likelihood on the process being i.i.d. can by computed by:

$$likelihood = 2P/m^2$$

4. Step 1: Assume $G_1$ and $G_2$ are SCC's, respectively. Let $M_1$ be the adjacent matrix of $G_1$ and $M_2$ be the adjacency matrix of $G_2$.

   Step 2: We denote the largest eigenvalues of $M_1$ and $M_2$ are $\lambda_1$ and $\lambda_2$, also known as the Perron number.

   Step 3: $M_1^n$ represents the total number of walks with length $n$ in $M_1$, which can be approximated by

$$M_1^n = \lambda_1^n * v_{\lambda_1} * u_{\lambda_1}^T$$

   where $\lambda_1$ is the Perron number of $M_1$, $v_{\lambda_1}$is the left eigenvector of $\lambda_1$, $u_{\lambda_1}$is the right eigenvector of $\lambda_1$.

Step 4: In $G_1$, the total number of walks from node $i$ to node $j$ (in original question, they are $v_1$ and $u_1$) with length $n$, can be approximated by

$$M_1^n[i,j] = \frac{v_i * u_j}{||u||} * \lambda_1^n * v_{\lambda_1} * u_{\lambda_1}^T$$

where $||u|| = \sum_k u_k$, and $v_i$, $u_j$ are the components in vectors $v$, $u$.

Step 5: Sum up all $M_1^n[i,j]$ for every length $n$, $i$ to node $j$ with the length less equal to $n$ is $S_1$,

$$S_1 = \sum_{n=1}^{n} M_1^n[i,j]$$

Step 6: In the same way, we can get $S_2$ for the total number of walks from node $a$ to node $b$ with the length less equal to $n$ in the $G_2$.

Step 7: If $S_1 - S_2 > 0$, we can decide the number of paths from $i$ to $j$ in $G_1$ is "more than" the number of paths from node $a$ to node $b$ in $G_2$ with the length less equal to $n$.