CSCE 689: Special Topics in Algorithms for Big Data	Fall 2023

Assignment 1 — 18 Sept, 2023

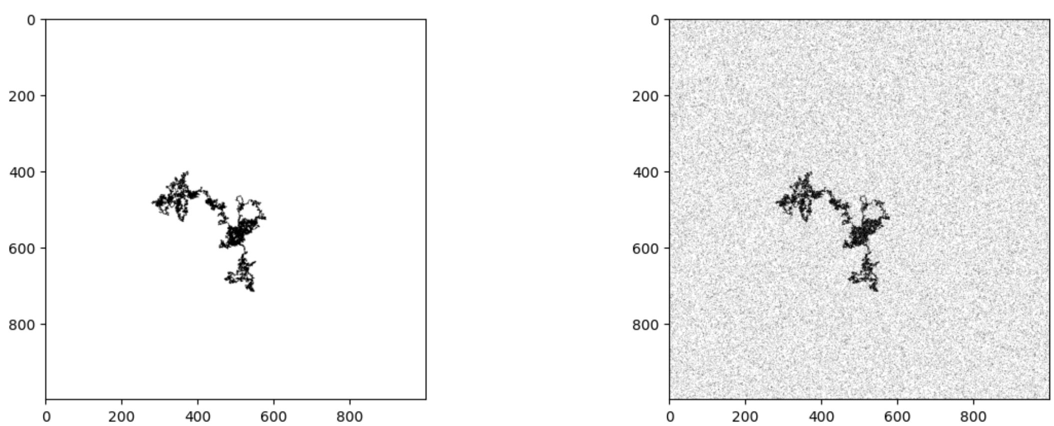
Prof. Victoria Crawford *Assignment: Shuo Xing*

Overview The project that I was assigned is to test whether the black pixels in an image form a connected shape. Let M be the matrix of an image I of size $n \times n$, and $M(i,j)$ represents the value of the entry at the i-th row and j-th column. And $\forall i,j \in [n],\ M(i,j) \in \{0,1\}$ with 0 representing white pixels and 1 representing black pixels. The connectivity testing of images is accomplished by employing the algorithms T_3 and T_4 in [1]. And both of algorithms T_3 and T_4 would return *No* when the image is ϵ-far from connectivity, with probability of at least 2/3, i.e., $\delta = 2/3$. All the code related to this assignment has been released as open-source and is accessible in this repository.

Data generation The image which has the connectivity property are generated through random walk initiated from a randomly selected black pixel by the following steps.

- Generate an $n \times n$ empty (all entries are 0) matrix M.
- Randomly choose a pixel (a (i,j) pair), and let $M(i,j) = 1$.
- Randomly select the next pixel $(i + di, j + di)$ with $(di, dj) \in \{(0,1), (0,-1), (1,0), (-1,0)\}$, and let $M(i + di, j + di) = 1$.
- Repeat the above process k times, and every pixel can be repeatedly visited.

Then we flip each entry in the image matrix M with probability q, where if $q = 0$ then the image has the connectivity property and as q gets higher (up to a certain point) the image would get further away from having the connectivity property. The example of the randomly generated image and the corresponding flipped image, with $k = 50 \times n$ and $q = 0.1$, can be found in Figure 1.

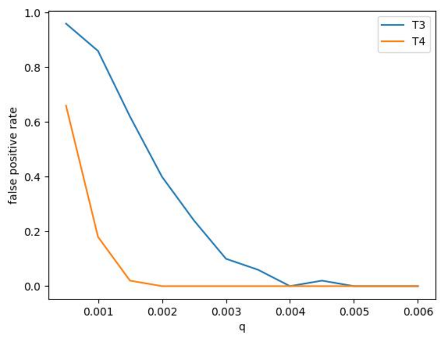


(a) Randomly generated image. (b) Flipped image.

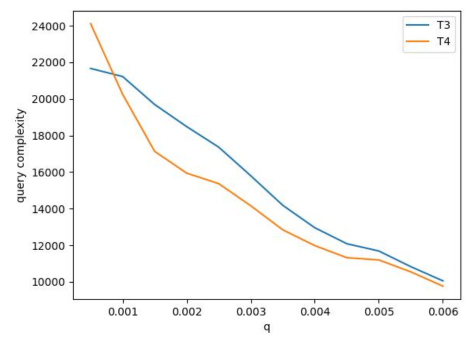
Figure 1: An example of randomly generated 1000×1000 image with $k = 50 \times n$ and the corresponding flipped image with $q = 0.1$.

Results We implement the algorithm T_3 and T_4 on 50 flipped images of size 1000×1000 (after generating random images by 50×1000 steps random walk) to test whether the flipped images are ϵ-far from connectivity property, obtaining results for $\epsilon = 0.1$, $\epsilon = 0.15$ and $\epsilon = 0.3$ respectively, as shown in Figures 2, 3 and 4.

For $\epsilon = 0.1$, $\epsilon = 0.15$ and $\epsilon = 0.3$, the false positive rate and average query times decrease as the q increases. This trend is expected because that the images would be further from connectivity property with increased q, leading to more accurate returned results and higher probability that algorithms can find the small components earlier.

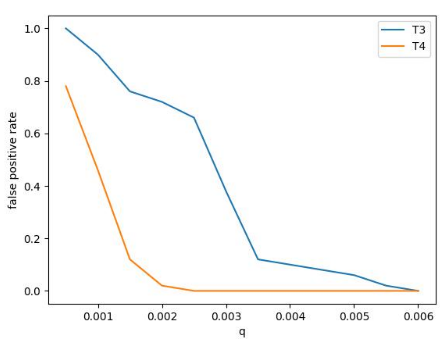


(a) False positive rate.

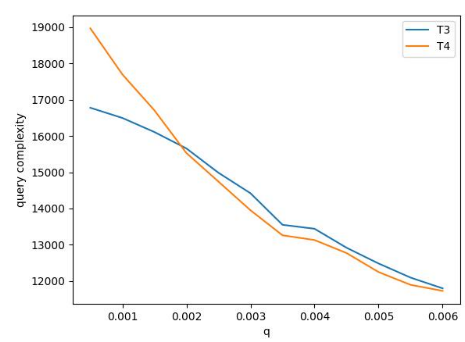


(b) Average query complexity.

Figure 2: Results of algorithms T_3 and T_4 with $\epsilon = 0.1$.



(a) False positive rate.

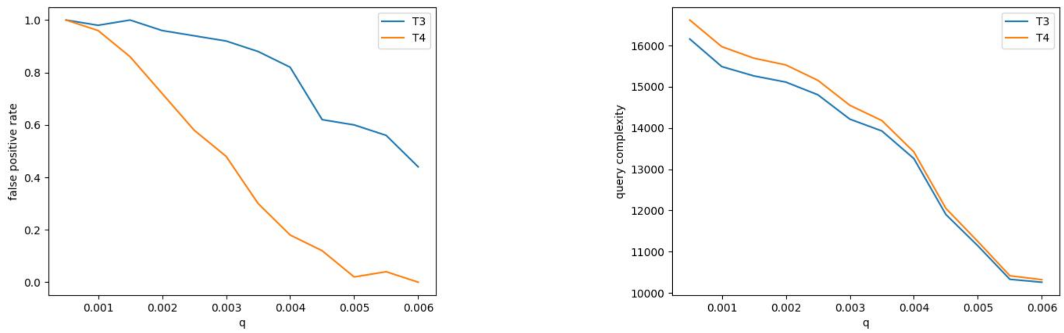


(b) Average query complexity.

Figure 3: Results of algorithms T_3 and T_4 with $\epsilon = 0.15$.

As the value of ϵ grows, the false positive rate of algorithms T_3 and T_4 would decrease across $q \in \{0.0005, 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035, 0.004, 0.0045, 0.005, 0.0055, 0.006\}$. This discrepancy arises from that the algorithm would query more pixels when the value of the ϵ is smaller, resulting in higher accuracy. And it also is the reason that the average query complexity of both algorithms T_3 and T_4 is higher when the value of ϵ is small.

As shown in Figures 2a, 3a and 4a, the false positive rate of algorithm T_3 consistently exceeds that of algorithm T_4 across different values of q. This is because that algorithm T_3 conducts the small



(a) False positive rate.

(b) Average query complexity.

Figure 4: Results of algorithms T_3 and T_4 with $\epsilon = 0.3$.

components searches $O(1/\epsilon^2)$ rounds, whereas algorithm T_4 conducts the search $O(1/\epsilon^2 \cdot \log(1/\epsilon^2))$ rounds, resulting in the higher probability algorithms T_4 can find small components and the black pixels outside the searching section.

As illustrated in Figures 2b and 3b, algorithm T_4 outperforms algorithm T_3 in terms of average query complexity. This trend aligns with expectations since the theoretical guarantee for query times in algorithm T_4 is $O(1/\epsilon^2 \cdot \log^2(1/\epsilon)$, which is better than the $O(1/\epsilon^4)$ guarantee provided by algorithm T_3. However, when $\epsilon = 0.3$, the average query complexity of algorithm T_4 is slightly higher than algorithm T_3 across various values of q. Despite it is not intuitive, such outcome could be possible. When the ϵ increases to 0.3, both algorithms T_3 and T_4 would sample fewer pixels to conduct components searches. And we have already known that algorithm T_4 performs more components searches than algorithm T_3 as the above analysis. Since the majority of the pixels in the flipped images are white, a significant portion of the sampled pixels for algorithms T_3 and T_4 would be white pixels. Therefore, the algorithm T_3 would performs much fewer breath-first searches than algorithm T_4, which might result in lower complexity for algorithm T_3. And this discrepancy becomes particularly evident when the value of q is small (the fraction of black pixels are small).

In conclusion, algorithm T_4 is generally more effective and efficient than algorithm T_3, although the average query complexity of algorithm T_4 might slight higher than algorithm T_3 in some cases.

References

[1] Sofya Raskhodnikova. Approximate Testing of Visual Properties. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 370–381. Springer, Berlin, Germany, 2003. doi:10.1007/978-3-540-45198-3_31.