Journal of educational equipment: Applied research, Volume 2, Issue 323 (September 2024)
ISSN 1859 - 0810

Thuật toán tăng trưởng ngẫu nhiên cho bài toán lấy mẫu ngẫu nhiên trong hình học tính toán

Nguyễn Khắc Quốc, Ngô Thanh Huy**
**Khoa Kỹ thuật và Công nghệ; Trường Đại học Trà Vinh*
Received: 16/8/2024; Accepted: 26/8/2024; Published: 06/9/2024

Abstract: *Computational geometry often suffers from problems of complexity and computing efficiency, particularly in multidimensional space. Situated in a geometric context, the paper focuses on random sampling algorithms for geometric optimization, including convex hulls, Delaunay triangles, and Voronoi diagrams. Using a random sample incremental technique, the author optimize the partition of space to represent a set of points in two- and three-dimensional space.*

Keyword: *Algorithms, randomized incremental, random sampling, computational geometry*

1. Đặt vấn đề

Vấn đề lấy mẫu ngẫu nhiên là một chủ đề cơ bản trong hình học tính toán với nhiều ứng dụng, từ hệ thống thông tin địa lý đến đồ họa máy tính. Để giải quyết hiệu quả vấn đề lấy mẫu ngẫu nhiên, một thuật toán tăng trưởng ngẫu nhiên được trình bày trong nghiên cứu này. Các thuật toán ngẫu nhiên cung cấp các đảm bảo xác suất có thể dẫn đến các giải pháp hiệu quả hơn. Công trình trước đây trong lĩnh vực này bao gồm các thuật toán xây dựng bao lồi, tam giác phân chia Delaunay, tìm kiếm láng giềng gần nhất được sử dụng để đơn giản hóa vấn đề và cải thiện hiệu suất. Tuy nhiên, cần có một cách tiếp cận tổng quát hơn có thể áp dụng cho nhiều vấn đề hình học khác nhau.

2. Nội dung nghiên cứu

2.1. Thuật toán lấy mẫu ngẫu nhiên

2.1.1. Mô tả thuật toán

Thuật toán lấy mẫu ngẫu nhiên (*Random Sampling Algorithms - RSA*) là một nhóm các phương pháp sử dụng kỹ thuật ngẫu nhiên hóa để giải quyết các bài toán hình học. Một số thuật toán ngẫu nhiên kinh điển như: Thuật toán Las Vegas điển hình là thuật toán QuickSort với pivot được chọn ngẫu nhiên, hoặc thuật toán tìm kiếm cực đại, cực tiểu trong đồ thị. Thuật toán ngẫu nhiên luôn cho kết quả chính xác nhưng thời gian chạy có thể thay đổi [1].

2.1.2. Ưu điểm của thuật toán lấy mẫu ngẫu nhiên

Các thuật toán thường đơn giản và dễ hiểu, không yêu cầu kiến thức chuyên sâu về toán học; Hiệu quả tính toán cao, có thể cho kết quả nhanh hơn các phương pháp định lượng truyền thống; Có thể áp dụng cho nhiều loại bài toán và lĩnh vực khác nhau.

2.1.3. Hạn chế của thuật toán lấy mẫu ngẫu nhiên

Kết quả có thể chỉ mang tính xấp xỉ và độ chính xác phụ thuộc vào số lượng mẫu; Kết quả và thời gian chạy có thể thay đổi giữa các lần thực hiện khác nhau; Để đạt độ chính xác cao, đôi khi cần số lượng mẫu lớn, dẫn đến tăng chi phí tính toán.

2.1.4. Một số thuật toán điển hình

Trong hình học, Monte Carlo được dùng trong các bài toán như tích phân đa chiều hay ước lượng thể tích của các hình dạng phức tạp [2]. Thuật toán RANSAC (*RANdom SAmple Consensus*) là một thuật toán được sử dụng rộng rãi trong xử lý ảnh và thị giác máy tính [3]. RANSAC có thể được dùng để tìm đường thẳng hoặc đường cong trong tập hợp các điểm dữ liệu bị nhiễu [4].

Thuật toán lấy mẫu ngẫu nhiên tuần tự (*Sequential Random Sampling*). Đây là một kỹ thuật lấy mẫu nơi các điểm được chọn lần lượt từ tập dữ liệu. Mỗi điểm mới được chọn sao cho cải thiện dần độ chính xác của mô hình hoặc bài toán đang giải quyết [5]. Thuật toán lấy mẫu ngẫu nhiên có trọng số (*Weighted Random Sampling*). Trong một số trường hợp, các điểm trong không gian hình học có trọng số khác nhau và được chọn theo các trọng số này [6]. Thuật toán đo đạc hình học ngẫu nhiên (*Randomized Geometric Measurements*). Phương pháp này được dùng để đo lường các thuộc tính hình học của các hình dạng phức tạp thông qua các phép đo ngẫu nhiên [7].

2.2. Thuật toán tăng trưởng ngẫu nhiên cho hình học tính toán

2.2.1. Mô tả thuật toán

Thuật toán tăng trưởng ngẫu nhiên là một kỹ thuật thường được sử dụng trong hình học tính toán để tạo ra các cấu trúc hoặc mô hình từ một điểm ban đầu, bằng cách thêm các phần tử mới vào cấu trúc hiện có theo cách ngẫu nhiên nhưng có kiểm soát. Các bước cơ bản của thuật toán tăng trưởng ngẫu nhiên:

Khởi tạo: Bắt đầu với một điểm hoặc một tập hợp điểm ban đầu là hạt nhân của sự tăng trưởng.

Vùng phát triển: Xác định vùng không gian trong đó sự tăng trưởng có thể xảy ra.

Thêm phần tử mới: Chọn vị trí ngẫu nhiên; Kiểm tra điều kiện. Các điều kiện này bao gồm khoảng cách tới các phần tử hiện có, hình dạng của cấu trúc hoặc các ràng buộc vật lý.

Lặp lại: Lặp lại quá trình chọn vị trí và thêm phần tử mới đến khi đạt được cấu trúc mong muốn.

2.2.2. Phân tích và đánh giá thuật toán

Giả sử cho một tập *S* với *n* số và tìm câu trả lời *những truy vấn lân cận*: một truy vấn sẽ là một số và có hay không số đó là lân cận của *S*. Chúng ta đặt một mẫu *R* ngẫu nhiên của *r* số trong tập *S*, với *r* là hằng số được chọn. Sắp xếp các phần tử của *R* (trong một hằng thời gian) và sau đó phân hoạch *S**R* (trong thời gian O) vào tập con *r* + 1; ở bước thứ *i* tập con chứa các phần tử này của *S**R* là chắc chắn lớn hơn *i* phần tử của *R*. Gọi mẫu *R* là tốt nếu với mỗi kết quả của tập con *r* + 1 của *S**R* có kích cỡ (*an* log *r*)/*r*, xác định hằng số *a* thích hợp. Cho một mẫu *R*, có thể kiểm tra nó có tốt hay không trong thời gian O. Thật vậy, trong thời gian O có thể tìm ra một mẫu tốt (bằng việc lặp lại tiến trình lấy mẫu bất cứ khi nào cũng cho được mẫu là không tốt). Với mỗi tập con chứa nhiều hơn *b* phần tử, giá trị hằng số $b > r$, chọn ngẫu nhiên lại một tập con của *r* phần tử từ nó và cứ như thế. Tiến trình này suy ra kiểu tìm kiếm tự nhiên trên cây và tiến trình tìm kiếm cho một truy vấn là rõ ràng. Cho truy vấn *q*, đồng nhất một trong những tập con *r* + *1* của *S* và tiếp tục tìm kiếm *q* và tìm kiếm lại trên cây con kết hợp với tập con này. Gọi Q là đánh giá của việc tìm kiếm trong một tập chứa *n* phần tử. Chúng ta có:

$$Q(n) \leq c + Q\left(\frac{an\, log\, log\, r}{r}\right)$$

Với *a* là nhỏ so với *r* /log*r* và *c* là một hằng số biểu diễn bậc giảm dần của cây. Dễ dàng nhìn thấy kết quả là Q = O(log*n*).

Việc sắp xếp các điểm cục bộ: Gọi *L* là tập *n* đường trong mặt phẳng. Các đường trong *L* phân hoạch mặt phẳng thành O(n^2) miền đa giác lồi là không có đường biên. Ở đây chúng ta chỉ quan tâm đến việc sắp xếp của các đường trong tam giác τ cố định, chứa phần trong tất cả các điểm của giao giữa các đường trong *L*. Với mỗi điểm tại vị trí mà 2 đường gặp nhau là đỉnh của đồ thị (phần dư này không có 3 đường của *L* gặp nhau tại một điểm). Hơn nữa, cũng có một đỉnh cho mỗi điểm tại vị trí mà một đường của *L* giao với đường biên của τ. Một cạnh giữa 2 đỉnh tương ứng trong phương tự nhiên, đến một đoạn giữa 2 đỉnh lân cận trong việc sắp xếp này. Mỗi mặt của đồ thị phẳng này là một trong các miền thuộc đa giác τ là đã được phân hoạch bởi các đường trong *L*.

Cho điểm truy vấn *q* trong mặt phẳng, mỗi mặt của đồ thị phẳng sẽ là một tam giác phân. Chúng ta sẽ qui nó về như là việc sắp xếp tam giác của các đường trong *L* và nó được chứa bởi ***T***. Khi cấu trúc hình học có độ chính xác là ***T*** phụ thuộc vào tam giác lớn bên trong, nơi chứa các điểm giao của các đường trong *L*. Tuy nhiên, số lượng xuất hiện của mỗi vùng với lý do sau:

(i) Bài toán điểm cục bộ, đồng nhất các mặt bên trong nơi mà một điểm không bị ảnh hưởng bởi việc chọn của đường biên tam giác, chấp nhận hình dạng biến thiên của các mặt.

(ii) Cho hầu hết các phần chứa tam giác sẽ bị ẩn và duy nhất.

Trong bài toán điểm cục bộ việc sắp xếp các tam giác phân của các đường ***T***. Thuật toán và cấu trúc dữ liệu như sau:

(i) Chọn mẫu ngẫu nhiên *R* của các đường *r* từ *L*, *r* thích hợp là một hằng số lớn. Xây dựng cách sắp xếp ***T***(*R*), số các mặt trong ***T***(*R*) là O(r^2), như là một hằng số.

(ii) Mỗi mặt *f* trong ***T***(*R*) xác định tập các đường của *L**R* giao với *f*, tập các đường này chứa L_f. Thuật toán có thể chạy trong O(nr^2). Với mỗi mặt *f* của ***T***(*R* với $|L_f| > b$, *b* là một hằng số. Thuật toán lặp lại tiến trình này, ghi lại các bước và duy trì một con trỏ từ mỗi mặt *f* .

Phân tích thời gian chạy của cấu trúc thủ tục trên và giải thích tiến trình tìm kiếm. Cho một điểm truy vấn *q*, chúng ta xác định (trong thời gian O(*1*)) một mặt *f* của ***T***(*R*) chứa *q*.

Chúng ta tiếp tục gọi lại việc tìm kiếm trong phạm vi $\boldsymbol{T}(L_f)$.

Chúng ta biết rằng $|L_f| \leq (an\log\log r)/r$ tìm trực tiếp trong thời gian Q. Vì vậy Q = O(log *n*).

2.2.3. Chứng minh

Gọi *S* chứa một tập tất cả các điểm, một đường của *L* giao với đường biên tam giác. Gọi Δ chứa một tập tất cả bộ 3 điểm từ *S*. Xác suất để một tam giác được xác định bởi một bộ 3 từ Δ xảy ra trong $\boldsymbol{T}(R)$ và được giao bởi nhiều hơn (*an* log*r*)/*r* đường của *L*.

Cho bộ 3 δ ∈ Δ, gọi *I*(δ) chứa một tập các đường của *L* giao với tam giác xảy ra bởi δ. Gọi *G*(δ) chứa các đường của *L* dạng các điểm trong δ (rõ ràng). Cận xác suất một bộ 3 δ xác định một mặt của $\boldsymbol{T}(R)$, chúng ta ghi ra kết quả của hai xác suất sau: Gọi $\varepsilon_1(\delta)$ bao hàm ngay cả các đường của *G*(δ) là trong *R*, và $\varepsilon_2(\delta)$ bao hàm ngay cả không có các đường *I*(δ) trong *R*. Rõ ràng, cả hai $\varepsilon_1(\delta)$ và $\varepsilon_2(\delta)$ phải gọi lại trong trật tự δ để xác định một mặt của *R*. Khi đó:

Xuất hiện $Pr[\delta]$ một mặt của

$T(R) \leq Pr[\varepsilon_1(\delta)]Pr[\varepsilon_2(\delta)|\varepsilon_1(\delta)]$

Bây giờ đường biên $Pr[\varepsilon_2(\delta)|\varepsilon_1(\delta)]$ đã được chọn các đường trong *G*(δ). Chúng ta quan tâm đến cái gì đã xảy ra trong sự còn lại các đường vẽ *r* − |*G*(δ)| của *R*. Đặc biệt, quan tâm là xác suất không còn lại đường vẽ *r* − |*G*(δ)| nào, chọn bất kỳ đường nào trong *I*(*δ*). Cận này là:

$$\prod_{i=0}^{r-|G(\delta)-1|}\left(1-\frac{|I(\delta)|}{n-|G(\delta)|-i}\right) \leq \left(1-\frac{|I(\delta)|}{n}\right)^{r-|G(\delta)|} \leq e^{-rI(\delta)/2n}$$

Cho bất kỳ giá trị *r* > 12 (khi |*G*(δ)| ≤ 6). Chúng ta chỉ giao được trong δ như $I(\delta) > (an\log\log r)/r$; gọi là các bộ 3 lớn. Thật vậy, với các bộ 3 lớn chúng ta có $Pr[\varepsilon_2(\delta)| < r^{a/2}]$. Khi đó:

Pr một bộ 3 lớn xuất hiện như một mặt của $T[(R)] \leq r^{-a/2}\sum Pr[\varepsilon_1(\delta)]$ các bộ 3 lớn.

Tóm lại, trong phần này kỳ vọng của các bộ 3 lớn chắc chắn trong *R*. Ngay khi *R* là một sự sắp xếp các đường *r*, và với mỗi điểm của các bộ 3 là được tạo thành bởi hầu hết 2 đường, hệ quả sau chỉ ra tổng này không bao giờ lớn hơn r^6, khi với *a* > 12. Như vậy, thời gian xây dựng đủ để suy ra:

$$T(n) \leq n^2 + cr^2T\left(\frac{an\log\log r}{r}\right)$$

Với *c* là một hằng số và *T*(*k*) chứa đường biên trên trong kỳ vọng đánh giá của việc xây dựng cấu trúc dữ liệu sắp xếp của *k* đường. Kết quả giải ra ta được $T = O(n^{2+\varepsilon(r)})$ với ε(*r*) là một hằng số xác định trở nên nhỏ hơn khi *r* lớn hơn.

3. Kết luận

Thuật toán tăng trưởng ngẫu nhiên được trình bày trong bài báo, cung cấp một giải pháp hiệu quả cho vấn đề lấy mẫu ngẫu nhiên trong hình học tính toán. Nhờ tính đơn giản, hiệu quả và khả năng mở rộng, thuật toán này trở thành một công cụ hữu ích cho nhiều ứng dụng hình học khác nhau. Thuật toán cải thiện hiệu suất và độ chính xác, vượt trội so với các phương pháp truyền thống. Khả năng linh hoạt và thích ứng của nó cho phép áp dụng vào các vấn đề hình học phức tạp hơn và mở rộng sang các lĩnh vực khác, cung cấp các giải pháp hiệu quả và đáng tin cậy.

Tài liệu tham khảo

[1] Rajeev Motwani, Prabhakar Raghavan, *Randomized Algorithms*, Stanford University, 2012.

[2] S. K. Mangla, P. Kumar, M. K. Barua, *Monte Carlo Simulation Based Approach to Manage Risks in Operational Networks in Green Supply Chain* Procedia Eng., vol. 97, pp. 2186–2194, 2014.

[3] WEI, Tong, et al. *Generalized differentiable RANSAC*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17649-17660, 2023

[4] Martínez-Otzeta, José María, et al. *Ransac for robotic applications: A survey,* 2022.

[5] Shekelyan, M., & Cormode, G. Sequential random sampling revisited: hidden shuffle method. In *International Conference on Artificial Intelligence and Statistics,* pp. 3628-3636. PMLR, 2021.

[6] Lehmann, Hans-Peter, *Weighted Random Sampling*, 2020.

[7] IMAI, Satoya. *Randomized measurements as a tool in quantum information processing,* 2023.

[8] Clarkson, K. L., & Shor, P. W., *Applications of random sampling in computational geometry*, II. Discrete & Computational Geometry, pp. 387-421, 1989.

[9] Lagae, A., & Dutr'e, P., *A comparison of methods for generating Poisson disk distributions*, Computer Graphics Forum, pp. 114-129, 2008.