

## Chương 3: Thuật toán Vegas

### 3.1 Giới thiệu

Phương pháp Monte Carlo sử dụng các thuật toán để giải quyết các bài toán trên máy tính bằng cách lấy mẫu ngẫu nhiên. Kết quả của phương pháp Monte Carlo này càng chính xác khi số lượng mẫu càng tăng. Theo quy luật số lớn khi kích thước mẫu càng lớn thì tốc độ hội tụ của tích phân càng lớn. Nhưng khi mở rộng số chiều của tích phân thì số lượng mẫu cũng phải tăng theo. Điều này cũng làm ảnh hưởng đến tốc độ tính toán của máy tính. Nếu cố định số lượng mẫu thì tốc độ hội tụ của tích phân sẽ giảm khi số chiều tăng lên. Nhìn chung, tốc độ hội tụ của phương pháp Monte Carlo là khá thấp. Vì vậy, phương pháp tích phân Monte-Carlo sử dụng thuật toán Vegas sẽ giải quyết hiệu quả với các tích phân nhiều chiều.

Đặc điểm thuật toán:

1. Phương sai chính xác cho tích phân thì được tính toán dễ dàng.
2. Hàm tính tích phân không cần liên tục khi sử dụng thuật toán mới này. Đặc biệt, các hàm bước nhảy cũng không có vấn đề gì. Vì thế để tính tích phân nhiều chiều là việc đơn giản.
3. Tốc độ hội tụ không phụ thuộc vào số chiều của tích phân.
4. Đây là thuật toán thích nghi, nó tự động tập trung tính toán tích phân tại những vùng có tích phân quan trọng nhất.

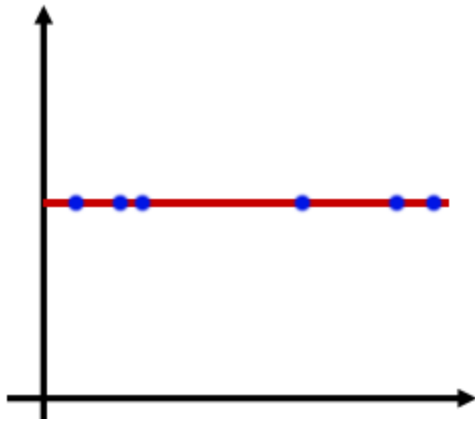
Đặc điểm (1) và (3) là phổ biến trong tất cả phương pháp Monte Carlo. Với (4) là một tính năng quan trọng nhất trong thuật toán này. Vấn đề chính trong tính toán các tích phân nhiều chiều là việc tăng số lượng mẫu theo cấp số nhân khi tăng số chiều. Do đó, mục đích chung của thuật toán để tính toán tích phân nhiều chiều là thích nghi.

### 3.2 Tích phân Monte Carlo sử dụng thuật toán Vegas

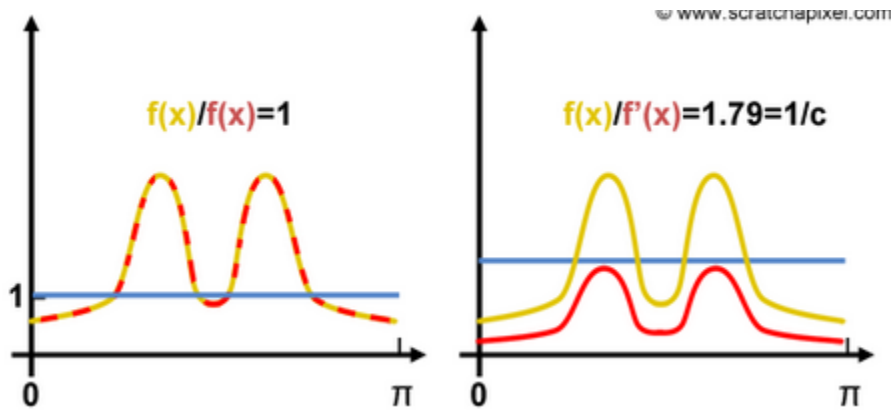
Monte Carlo thật ra là một phương pháp khá đơn giản. Nhưng vấn đề giảm phương sai cần được quan tâm. Để giảm phương sai, tất cả những gì có thể làm là tăng số lượng mẫu  $N$ . Muốn giảm phương sai xuống hai lần thì phải tăng gấp bốn lần số mẫu. Đối với các hàm tích phân nhiều chiều, số lượng mẫu tăng theo cấp số nhân khi số chiều tăng lên. Do đó, có nhiều phương pháp đã được nghiên cứu để giảm phương sai mà không cần tăng số lượng mẫu  $N$ . Như vậy, phương pháp lấy mẫu trọng và phương pháp lấy mẫu phân tầng được thuật toán Vegas sử dụng để giảm phương sai.

#### 3.2.1 Lấy mẫu quan trọng

Lấy mẫu trọng là phương pháp làm giảm phương sai bằng cách chọn hàm mật độ xác suất (PDF) bất kỳ. Nhưng nếu chọn hàm mật độ  $pdf(x)$  tương tự với hình dạng của hàm  $f(x)$  thì kết quả tích phân càng chính xác. Nhận thấy rằng nếu đặt nhiều mẫu tại những nơi mà tích phân có đóng góp cao. Thì phương sai của tích phân MC giảm đáng kể.



Xét hàm lấy tích phân là một hàm hằng (hình 4). Với hàm phân phối xác đều. Thì kết quả tích sẽ rất chính xác. Bởi vì vị trí mẫu không làm ảnh hưởng đến kết quả tích phân. Hay nói cách khác là phương sai bằng 0. Mục đích chính của phương pháp tích phân Monte Carlo là giảm phương sai đến mức thấp nhất. Nhưng thật không may, trong thực tế rất hiếm gặp hàm lấy tích phân là hàm hằng. Thậm chí, các hàm còn có đỉnh cao và hẹp. Như vậy, chúng ta có thể biến đổi một hàm không phải là hàm hằng thành hàm hằng hay không. Điều này là hoàn toàn có thể khi chia  $f(\mathbf{X}) / f'(\mathbf{X})$ . Với  $f(\mathbf{X})$  là giá trị tại một biến ngẫu nhiên và  $f'(\mathbf{X})$  là kết quả chính xác. Ví dụ,  $f(X) = 2$ , tại  $X=0$  thì  $f(0)/2=1$  và  $f(X)=0.5$  tại  $X=2$  thì  $f(2)/0.5=1$ . Nếu thực hiện như vậy với tất cả các biến ngẫu nhiên. Cuối cùng, bài toán sẽ có được hàm phân bố xác suất là hàm hằng.



Như trên hình, Giả sử  $f(X)$  là hàm cần tính tích phân,  $f'(X)$  là hàm phân phối xác suất. Kết quả tích phân sẽ không đổi khi lấy  $f(X)/f'(X)$ . Với điều kiện  $f(X)=cf'(X)$

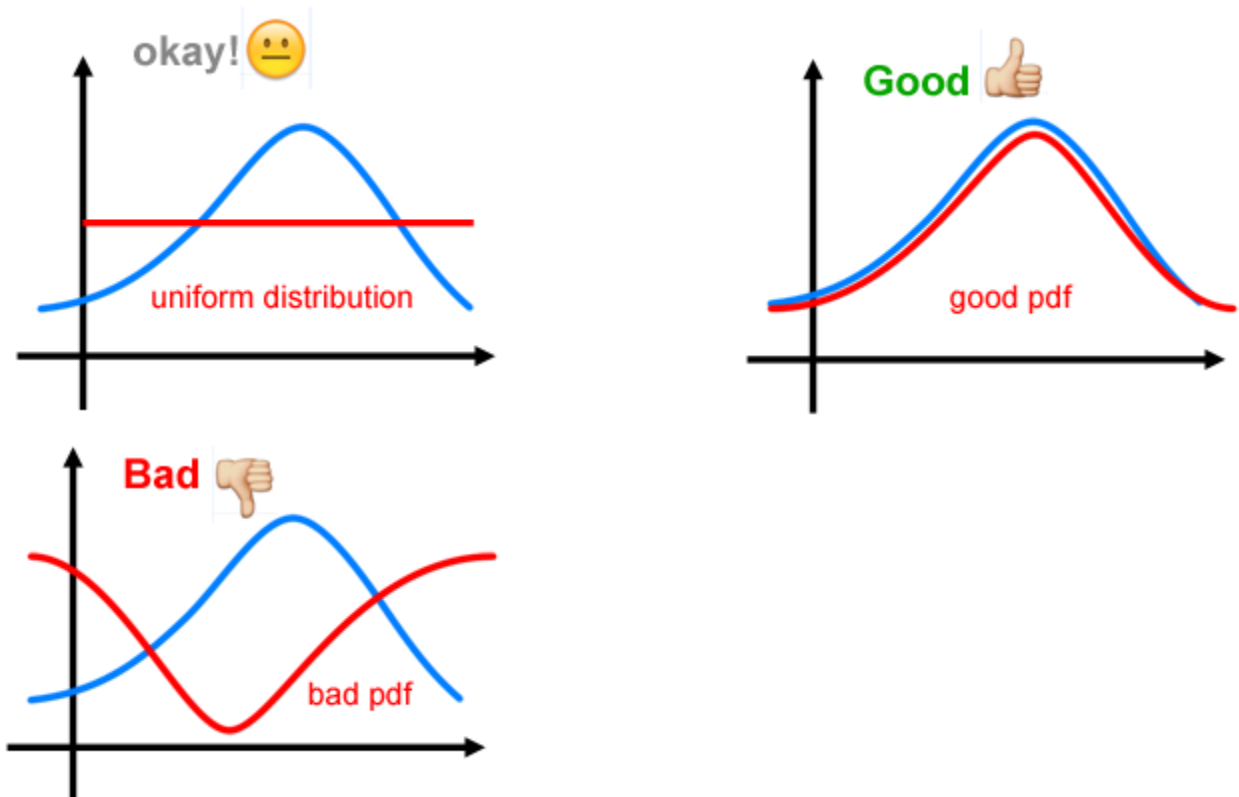
Và  $f(X)/f'(X) = 1/c$

$$\langle F^N \rangle = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(x)}{pdf(x)}.$$

Vậy tích phân Monte Carlo:

Như vậy:  $pdf'(X) = cf(X)$

Nhìn chúng, nếu hình dạng hàm pdf'(X) giống hình dạng của f(X) thì phương sai của tích phân Monte Carlo sẽ bằng 0. Đây cũng là hạn chế của phương pháp lấy mẫu trọng bởi vì phương pháp này phụ thuộc vào việc lựa chọn hàm PDF cho bài toán.



### 3.2.2 Lấy mẫu phân tầng.

Để giảm phương sai, thể tích tích phân có thể được chia thành  $N$  khối thể tích nhỏ với các kích thước khác nhau. Sau đó tích phân Monte-carlo được thực hiện trong mỗi khối thể tích nhỏ. Với mỗi khối thể tích nhỏ này sử dụng  $M/N$  điểm ngẫu nhiên. Khi kích thước của khối thể tích nhỏ bị thay đổi thì phương sai sẽ được tính toán lại và được tối ưu khi phương sai của mỗi khối thể tích nhỏ này là như nhau ( $= \sigma^2/N$ ). Do đó, khi sử dụng gieo điểm phân tầng, tích phân sẽ được tập trung tại nơi có sai số lớn nhất, tức là nơi tích phân vừa lớn và vừa dao động mạnh.

Thuật toán sử dụng phương pháp lấy mẫu phân tầng. Ban đầu thuật toán chia khối thể tích của tích phân  $n$  chiều thành  $N^n$  khối nhỏ như nhau. Bằng cách sử dụng một lưới hình chữ nhật đồng nhất. Mỗi khối nhỏ sử dụng hai điểm để đóng góp vào việc tính tích phân Monte Carlo và phương sai. Các phương sai từ các khối nhỏ sau đó được sử dụng để định nghĩa lại không gian lưới mới dọc

theo mỗi trục. Và không gian lưới mới này sẽ được sử dụng cho lần lặp tiếp theo. Và phải giữ cho tổng số khối nhỏ không đổi. Như vậy qua mỗi vòng lặp thì các khối nhỏ có thể tập trung dần dần về nơi có phương sai lớn nhất trước đó. Như vậy phương sai được giảm. Thuật toán này được sử dụng rộng rãi trong ngành Vật lý Lý thuyết và rất thành công cho nhiều ứng dụng tính toán hai chiều trở lên. Sức mạnh lớn nhất của phương pháp là khả năng thích nghi của tích phân đang được xem xét. Tuy nhiên khả năng thích nghi của nó được quyết định bởi số lượng đoạn được chia dọc theo mỗi trục (N). Và N bị giới hạn bởi tổng số tích phân (M) được cho phép ở mỗi vòng lặp.

$$M = 2N^n$$

Hạn chế này cho thấy một khuyết điểm lớn khi có số chiều lớn ( $M < 10^6$  khi  $n = 9$  và  $N \leq 4$ ).

### 3.2.3 Thuật toán Vegas

Hạn chế từ phương pháp lấy mẫu phân tầng có thể tránh được bằng cách sử dụng phương pháp lấy mẫu trọng thay vì phương pháp lấy mẫu phân tầng để giảm phương sai. Phương pháp lấy mẫu quan trọng dường như cho thấy kém hiệu quả hơn phương pháp lấy mẫu phân tầng. Tuy nhiên trong thực tế khả năng thích nghi vượt trội hơn khi được thể hiện ở bài toán có số chiều lớn. Vì vậy thuật toán Vegas sẽ sử dụng cả hai phương pháp này để xử lý bài toán.

**Xem xét tích phân một chiều:**

$$I = \int_0^1 dx f(x)$$

Khởi tạo M điểm với mật độ xác suất đều ( $p(x) = 1$ ). Ngoài việc ước tính giá trị tích phân (1) và phương sai (2), các điểm M dùng để tính tích phân cũng có thể được sử dụng để tính toán lại mật độ xác suất tốt hơn cho vòng lặp tiếp theo. Trong phương pháp này phương sai sẽ được giảm dần qua các vòng lặp.

Có một số kỹ thuật để tạo ra dãy số giả ngẫu nhiên với phân bố đều. Sẽ khó khăn hơn khi muốn tạo ra dãy số ngẫu nhiên với một phân bố bất kỳ có mật độ  $p(x)$ . Với bài toán hiện tại sẽ đơn giản hơn nếu ta coi  $p(x)$  là một hàm bước nhảy với N bước. Xác suất của một số ngẫu nhiên được chọn từ một bước nhảy bất kỳ thì được xác định là một hằng số  $1/N$  cho tất cả các bước ( $0 = x_0 < \dots < x_N = 1$ ,  $\Delta x_i = x_i - x_{i-1}$ )

$$p(x) = \frac{1}{N\Delta x_i}, \quad x_i - \Delta x_i \leq x < x_i, \quad i = 1, \dots, N,$$

$$\sum_{i=1}^N \Delta x_i = 1$$

Phân phối xác suất được điều chỉnh theo các tích phân từng phần cụ thể bằng cách thay đổi kích thước của các đoạn  $\Delta x_i$ . Trên thực tế thì N bị giới hạn bởi bộ nhớ máy tính và thường thì N là một hằng số (N = 50 đến 100).

Cho các M điểm tính tích phân, phân bố xác suất. Kích thước các đoạn  $\Delta x_i$  được điều chỉnh lại bằng cách chia nhỏ mỗi đoạn thành  $m_i + 1$  đoạn nhỏ hơn, tại

$$m_i = K \left( \frac{\bar{f}_i \Delta x_i}{\sum_j \bar{f}_j \Delta x_j} \right) \quad (4)$$

Và

$$\bar{f}_i \equiv \sum_{x \in x_i - \Delta x_i}^{x_i} |f(x)|$$

Toàn miền lấy tích phân sẽ bị chia nhỏ thành  $k+1$  đoạn con (thường thì  $k = 1000$ ). Số lượng đoạn nhỏ trong mỗi đoạn được xác định trong biểu thức (4). Mỗi đoạn nhỏ của mỗi đoạn lớn là như nhau. Sự đóng góp của mỗi đoạn cho hàm trọng sẽ tăng tỉ lệ với sự đóng góp của nó cho tích phân  $|f(x)|$ , như yêu cầu trong phương trình (3). Do cần tổng số đoạn vẫn giữ nguyên là N, các đoạn con sẽ được gom lại thành các đoạn mới sao cho số đoạn con trong mỗi nhóm là như nhau và không đổi (để giữ mật độ tỉ đối giữa các đoạn). Hệ quả là ta phải thay đổi kích thước của các đoạn nhưng giữ tổng số đoạn không đổi sao cho đoạn có kích thước nhỏ nhất ở nơi  $|f(x)|$  lớn nhất. Lưới mới này sẽ được dùng và sẽ được tinh chỉnh trong mỗi lần lặp cho tới khi ta có được lưới tối ưu (tức  $m_i = m_j$ ,  $i, j = 1, \dots, N$ ).

Trong thực tế, cách tốt nhất để làm giảm sự phân chia của thuật toán một cách nhanh chóng và mất ổn định khi thay đổi lưới từ vòng lặp này sang vòng lặp khác, ta sử dụng [5]

$$m_i = K \left\{ \left[ \frac{\bar{f}_i \Delta x_i}{\sum_j \bar{f}_j \Delta x_j} - 1 \right] \frac{1}{\log[\bar{f}_i \Delta x_i / \sum_j \bar{f}_j \Delta x_j]} \right\}^\alpha$$

Thay cho (4). Tham số  $\alpha$  xác định tốc độ hội tụ và thường nằm trong khoảng (1,2).

Các giá trị  $\bar{f}_i$  phải được xóa bỏ sau mỗi vòng lặp bởi vì bộ nhớ máy tính có giới hạn.

Tuy nhiên, các ước tính và sai số của tích phân có thể sử dụng cho tất cả các đánh giá của tích phân.

$$\begin{aligned} \bar{I} &= \sigma_I^2 \sum_i \frac{I_i}{\sigma_i^2} \\ \sigma_I &= \left[ \sum_i \frac{1}{\sigma_i^2} \right]^{-1/2}. \end{aligned} \quad (5)$$

$I_i$  và  $\sigma_i^2$  là tích phân và độ lệch chuẩn trong vòng lặp  $i$  bằng cách sử dụng hai phương trình (1) và (2). Khi tích phân có đỉnh cao và hẹp,  $I_i$  và  $\sigma_i^2$  thì ước tính tích phân đôi khi không chính xác trong những vòng lặp đầu tiên (trước khi tích phân có sự thích nghi). Và cách tốt nhất là thay thế phương trình (5) bằng

$$\begin{aligned}\bar{I} &= \sum_i I_i \left( \frac{I_i^2}{\sigma_i^2} \right) / \sum_i \frac{I_i^2}{\sigma_i^2}, \\ \sigma_I &= \bar{I} \left[ \sum_i \frac{I_i^2}{\sigma_i^2} \right]^{-1/2}.\end{aligned}\tag{6}$$

Trong mọi trường hợp  $X^2$  không được vượt quá số lần lặp. Thuật toán có thể không còn đúng khi thực hiện nó.

Số lặp và số lần tính tích phân cần cho mỗi lần lặp rõ ràng đều phụ thuộc vào độ phức tạp của hàm lấy tích phân và độ chính xác tìm được. Nói chung, tốt nhất để sử dụng như là một số đánh giá tính phân ở mỗi vòng lặp

### Xem xét tích phân n chiều

Thuật toán được mô tả bên dưới được tổng quát hóa với mục đích xử lý tích phân nhiều chiều. Để minh họa cho sự thay đổi

$$I = \int_0^1 dx \int_0^1 dy f(x, y).$$

Hàm mật độ xác suất được tách rời để phù hợp với bộ nhớ lưu trữ có giới hạn của máy tính.

$$p(x, y) = p_x(x) p_y(y).$$

Trong trường hợp này mật độ tối ưu được thể hiện với một kết quả tương tự cho  $P_y(y)$ .

$$p_x(x) = \left[ \int_0^1 dy \frac{f^2(x, y)}{p_y(y)} \right]^{1/2} / \int_0^1 dx \left[ \int_0^1 dy \frac{f^2(x, y)}{p_y(y)} \right]^{1/2}$$

Do đó thuật toán một chiều có thể áp dụng dọc theo một trục. Nhưng  $\bar{f}_i$  (phương trình (4)) được định nghĩa bằng

$$(\tilde{f}_i)^2 = \sum_{x \in x_i - \Delta x_i}^{x_i} \sum_y \frac{f^2(x,y)}{p_y{}^2(y)}$$

$$\propto \frac{1}{\Delta x_i} \int_{x_i-\Delta x_i}^{x_i} dx \int_0^1 dy \frac{f^2(x,y)}{p_\nu(y)}$$