

Model Chandrasekhara / Smoluchowskiego - dowolna kombinacja pudeł

Piotr Piękos

21 sierpnia 2019

W tym dokumencie omówiony jest najogólniejszy przypadek modelu pudełek. Zakładamy tutaj, że przejścia pomiędzy pudełkami tworzą łańcuch Markowa. Dodatkowo wierzchołki x są opisane dwoma parametrami:

- częstotliwością narodzin w danym wierzchołku a_{xN}
- średnią długością przebywania w danym wierzchołku a_{xS}

W każdym wierzchołku mającym częstotliwość narodzin większą od zera cząstki regularnie się "rodzą". W każdym wierzchołku po wejściu do niego (lub po narodzeniu w nim) losowana jest długość życia z rozkładu wykładniczego o zadanym dla wierzchołka parametrze. Po wyjściu cząstka przechodzi do następnego wierzchołka (lub umiera) zgodnie z macierzą przejść.

W dokumencie przedstawiony jest algorytm do symulacji, opis interfejsu oraz przykładowe ustawienia Łańcuchów wraz z trajektoriami.

1 Interfejs

Do opisu środowiska potrzebne są dwa pliki:

- nodes - opisujący intensywność narodzin i długość życia we wierzchołku.
- transitions - opisujący macierz przejść po wyjściu z wierzchołka

Format:

nodes jest to plik składający się z dwóch kolumn oddzielonych liczbami. Numer wiersza odpowiada numerowi wierzchołka (licząc od 1). Pierwsza kolumna to intensywność narodzin, druga to długość życia.

transitions to surowy plik z macierzą przejść, kolumny oddzielone są spacjami. **Pierwsza kolumna jest wyróżniona i oznacza przejście do wyróżnionego wierzchołka (śmierć)**

Przykładowe środowiska są zawarte w folderach

- kolko
- koniczynka

2 Opis algorytmu

Wynikiem algorytmu będzie tabela ze zdarzeniami, gdzie zdarzeniem jest na przykład narodzenie cząsteczki, albo przejście jej z jednego pudła do innego. Oprócz informacji o zdarzeniu wiersz zawiera też czas wystąpienia zdarzenia

Algorytm składa się z dwóch części

1. Narodziny cząsteczek
2. Podróżowanie

W **pierwszej** części generujemy narodzone cząsteczki - wchodzimy do każdego "rodzącego" wierzchołka, symulujemy wszystkie narodziny które będą miały miejsce w danym wierzchołku podczas ustalonego horyzontu czasu.

W **drugiej** części "chodzimy" cząstkami między wierzchołkami. Ustalamy wierzchołek, dla wszystkich cząstek które aktualnie w nim

przebywają losujemy czas życia oraz nowy wierzchołek. Nadpisujemy "obecny stan" i zapisujemy eventy do historii.

Taka realizacja algorytmu pozwala na wektorową realizację losowania czasów życia i przejść próbek co znacząco przyspiesza czas symulacji.

3 Algorytm

1. Narodziny cząsteczek

for x in rodzące wierzchołki:

Gen $N \sim Poiss(a_{xN}t)$

for $i = 1$ to N do Gen $U_i \sim U(0, t)$

Dodaj pary $(x, \text{czas powstania})$ do historii oraz aktualnych wydarzeń

"aktualne wydarzenia" oznaczają ostatnie wydarzenie z daną cząstką, jest ono nadpisywane i przechowuje informacje gdzie jest w danym "momencie" (nie jest to jednak robione chronologicznie z czasem)

2. Podróżowanie

$w = 1$, W - ilość wierzchołków

while najmniejszy czas z aktualnych wydarzeń $< t$:

weź cząstki aktualnie będące w w .

Zasymuluj ich czas życia (z parametrem a_{wN}) oraz nowy wierzchołek

zaktualizuj aktualne wydarzenia i dodaj zmienione rzeczy do historii.

$w := w \bmod W + 1$ ($w \neq 0$, bo 0 to wierzchołek z martwymi cząsteczkami)

historia zawiera wszystkie wydarzenia, teraz wystarczy je posortować i odpowiednio przetransponować żeby mieć ilość cząsteczek w danej chwili

4 Przykładowe trajektorie