Simulation normalisierter BPDL Prozesse

Boris Prochnau

Geboren am 22. Dezember 1989 in Tartu

17. Juli 2014

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Anton Bovier

Institut für Angewandte Mathematik

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Inhaltsverzeichnis

1 Einleitung

Hier steht eine Einleitung bestehend aus:
Ziel dieser Bachelorarbeit...
Vorstellung des Populationsmodells...
Näheres zur Simulation...
Näheres zum erwarteten Verhalten der Simulation...
Welche Möglichkeiten das Programm bieten soll...
Schließlich erwähne ich noch TSS Prozesse und dass ich auch dafür ein Programm geschrieben habe welches den besonders interessanten Wechsel dominanter Spezies durch Invasion simulieren kann.

2 Modell

Hier woran sich das von mir verwendete Modell orientiert [1] und das es sich an Darwins Evolutionslehre anlehnt.
Dann erwähne ich noch kurz dass für die von mir erwartete Simulation ein etwas anderes Modell simuliert.

2.1 Grundlagen

Hier führe ich das von mir verwendete Modell ein.
Anschließend (im selben Kapitel) erkläre ich dass wir das Modell zwar aus der Sicht des Individuums erstellt haben, jedoch durch Superposition die Ereignisse auf die Ebene der Merkmale anheben können und so die gesamte Population simulieren können.

2.2 BPDL Prozess

Hier beschreibe den aus dem Modell resultierenden stochastischen Prozess und in welchem Raum er lebt.
Dort werde ich auch den eigentlichen Generator des BPDL Prozesses erwähnen und den daraus für mich abgeleiteten Generator aufschreiben.
Lässt sich noch etwas wichtiges zum Prozess sagen?

3 Normalisierung und Eigenschaften des BPDL Prozesses

Hier werde ich erklären dass man von dem Prozess Eigenschaften erwarten kann die man anhand der Fitness erklären kann oder einen stabilen Zustand zu dem der Prozess konvergiert.
Um diese Eigenschaften besser untersuchen zu können und um sie deutlich zu machen weise ich darauf hin dass wir die Normalisierung des BPDL Prozesses nutzten.

3.1 Normalisierung des BPDL Prozesses

Hier nenne ich den von Silke angeführten Grund, dass wir die LPA Normalisierung einführen um die Betrachtungsebene auf die der Population und weg vom Individuum zu heben. Ich würde auch gerne betonen dass diese Methode sich besonders dafür eignet wenn viele Individuen sich auf wenige Merkmale verteilen und man möglichst gut deren erwartetes Verhalten beobachten kann. *(Hier tauchen Bilder auf!)*

3.2 Fitness

Hier würde ich zunächst gerne die Fitnessfunktion einführen und wie Martina mir empfohlen hat auch auf den Aufbau der Fitness eingehen und versuchen zu erläutern wieso die Berechnung der Fitness so aussieht wie sie es tut. Natürlich erwähne ich dabei dass sie die anfängliche Wachstumsrate eines mutierten Merkmals darstellt usw.

3.3 Equilibrium

Soll ich das Kapitel vielleicht lieber "stabiler Zustand" nennen? Loren meinte dass Equilibrium vielleicht nicht in der Bachelorarbeit auftauchen sollte.
Hier erkläre ich das deterministische Verhalten von Merkmalen wenn $K \to \infty$ (ohne Mutation) und dass wir in der Regel einen stabilen Zustand beobachten können in der sich die Populationsgröße nicht mehr ändert.
Schließlich beschreibe ich die stabilen Zustände im monomorphen, dimorphen Fall und wann diese angenommen werden.
(Hier könnte ich Bilder einfügen wenn ihr das für eine gute Idee haltet)

4 Algorithmus zur Simulation eines normalisierten BPDL Prozesses

Hier weiße ich darauf hin dass sich der Pseudocode von der Implementierung in so fern unterscheidet, dass bei der Implementierung sorgfältig darauf geachtet wurde die trennbaren Aufgaben nicht innerhalb einer Funktion auszuführen und jede Aufgabe einzeln ansprechbar ist.
Dies ist später für den Verhaltenstest entscheidend und auch aus weiteren Gründen sinnvoll die in "Aufgabenteilung und Flexibilität" angesprochen werden.

4.1 Implementierung

Hier stelle ich heuristisch die Implementierung eines Sprungs von ν_t vor.

4.2 Pseudocode

An dieser Stelle schreibe ich den gesamten Pseudocode auf und mache kleine Verweise auf die ausgelagerten Arbeitspakete die zuvor erwähnte Implementierung.

4.3 Optimierung für viele Merkmale

Hier werde ich die von mir schon besprochene Optimierung nennen welche bei vielen Merkmalen wichtig wird. Diese verwendet zur Berechnung der neuen Raten die alten Werte und korrigiert nur die alten Werte statt sie neu auszurechnen. Natürlich auch wie ich diese implementiert habe.

4.4 Aufwand

Ich bin noch nicht sicher ob ich diesen Teil erwähnen soll, aber ich möchte hier erwähnen dass z.B. bei den von mir gemessenen Werten die Berechnung der Todesrate die mit Abstand meiste Zeit in Anspruch nimmt und warum das klar ist. Schließlich vielleicht noch weitere Punkte die für den Aufwand einer solchen Simulation von Interesse sind, wie z.B. das Verhältnis von großen K, simulierter Zeit und notwendigen Sprüngen.

5 Das Programm

Hier sage ich vielleicht einige einleitende Worte zur Entwicklung des Programms.

5.1 Architektur: Aufgaben und Flexibilität

Hier werde ich einige Stolpersteine erwähnen denen ich versucht habe mit der von mir gewählten Architektur auszuweichen.
Dann führe ich anhand eines Diagramms 3 größere Arbeitspakete ein in die die Aufgaben einsortiert wurden und worauf geachtet wurde.
Vielleicht werde ich noch einige Vorteile meiner Vorgehensweise aufzählen oder etwas anderes was meine Konzepte betont.

5.2 Layout: Lesen und Darstellen der Parameter

Hier erkläre ich anhand von Screenshots meines Programms wie Parameter eingelesen und dargestellt werden. Damit ist auch die Fitness und die Berücksichtigung des "K" Parameters gemeint.

5.3 Layout: Erstellen neuer Instanzen

Wie erwartet erkläre ich hier wie eine neue Instanz erstellt wird (auch anhand von Bildern) und warum das in unserem Fall deutlich aufwendiger ist als es mit einer Konsole gewesen wäre.

5.4 Layout: Präsentation der Simulation

Zunächst nenne ich die verschiedenen Anforderungen an die Darstellung der Graphen (Anzeige, Zoom, Bewegungsfreiheit, Koordinatensystem, Speichermöglichkeit, Absturzsicherheit...).
Dann beschreibe ich ganz kurz wie man die Berechnung der Simulation startet und warum das in der Programmierung ein kritischer Punkt war.
Nach der Anzeige des Graphen beschreibe ich was nun tatsächlich Dargestellt wurde (Zeit und Größe, stabile Zustände, Anzahl der Sprünge, Legende mit Daten).
(Hier habe ich auch Bilder verwendet)

6 Verhaltenstest - Korrektheit der Implementation

Hier stelle ich das Konzept der "Testgetriebenen Entwicklung" das ich zuvor recherchiert und einstudiert habe. Ich erkläre dabei warum die zunehmende Komplexität des Programms den Code es notwendig macht solche Mittel zu verwenden und es sehr große Möglichkeiten nicht nur zur Kontrolle sondern auch zur Verhaltensanalyse bietet.

6.1 Unit Tests der Algorithmusmodule

Hier beabsichtige ich einige Tests und deren Resultate zu präsentieren um von der Korrektheit der Simulation zu überzeugen.

7 TSS Prozesse

Zunächst leite ich die TSS Prozesse ein und stelle die Änderung an der Mutation vor. An dieser Stelle sollte ich mich nochmal kurz mit einem von euch zusammen setzten um genauer zu besprechen warum diese Wahl der Mutation getroffen wurde.
Dann sage ich noch dass viele Sprünge um das Equilibrium zu erwarten sind und wie dieses Modell mit der gewöhnlichen BPDL Simulation aussehen würde. (Bilder)
Dann hebe ich noch einmal die Bedeutung der Fitness hervor und erkläre dass sich durch die Fitnessfunktion die Invasionswahrscheinlichkeit bestimmen lässt und wie ich das farblich hervorgehoben habe. Dabei sage ich noch kurz etwas zum Abbruchkriterium und zum verhalten wenn wir die Mutation nicht nur auf die Nachbarn beschränkt hätten.

7.1 Optimierung

Hier stelle ich die lineare Interpolation als eine Optimierung vor die Übersichtlichkeit, Analyse und Laufzeit deutlich verbessert. Dann beschreibe ich anhand von Bildern wie gut man Mutationen und deren Auswirkungen verfolgen kann und wie ich geprüft habe ob auch hier alles Korrekt läuft.
Schließlich erkläre ich wie ich die Mutationszeitpunkte ermittelt habe.

7.2 Algorithmus

Ich weiß nicht ob ich diesen Punkt einbringen soll, aber wenn dann werden hier die Details zum zusätzliche Aufwand wie zum Beispiel der Bestimmung eines Sprung Start und Endpunktes oder dem Abbruch erwähnt.

8 Ausblick

Hier werde ich wahrscheinlich einige Funktionen vermerken die es bis zu diesem Zeitpunkt nicht mehr in das Programm geschafft haben.

Literatur

[1] Nicolas Champagnat. A microscopic interpretation for adaptive dynamics trait substitution sequence models. *Stochastic Processes and their Applications*, 116(8):1127 – 1160, 2006.