Simulation normalisierter BPDL Prozesse

Boris Prochnau

Geboren am 22. Dezember 1989 in Tartu

11. 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

Das Ziel dieser Bachelorarbeit ist es eine Programm zu entwickeln welches den zeitlichen Verlauf von sogenannten normalisierten BPDL Prozessen graphisch darstellt.
Diese BPDL (von Bolker, Pacala, Dieckmann und Law) Prozesse basieren auf einem stochastischen Populationsmodell, welches im nächsten Kapitel näher vorgestellt wird.

Dem Programm sollte es möglich sein die Parameter einer Population festlegen zu können und die Entwicklung dieser Population schließlich zeitlich verfolgen zu können. Auf diese Weise soll man beobachten können ob sich ein Merkmal gegenüber einem anderen Durchsetzten kann oder sich ein stabiler Zustand einpendelt. Insbesondere kann der Einfluss der Normalisierung auf das vorzeitige Aussterben eines Prozesses deutlich dargestellt werden.

Alle Simulationen basieren auf einem Modell, dass jedes Lebewesen einer Population (z.B. Pflanzen) ein bestimmtes Merkmal trägt. Diese Merkmale werden durch Wettbewerb zu jeder existenten Gruppe, Geburten und Todesraten klassifiziert. Schließlich ist es jedoch die Entwicklung der Population und nicht der Individuen die simuliert werden soll, weshalb man im simulierten Prozess zwar den Tod und die Geburt von Individuen verfolgen kann, aber nicht die Entwicklung spezieller Individuen. Der Übergang zu dieser Sichtweise wird näher im 2. Kapitel beschrieben und krönt in der Konvergenz zu einer deterministischen Funktion.

Um zu Prüfen ob und wie schnell der Prozess gegen seinen Stabilen Zustand konvergiert, werden in der Simulation auch stabile Zustände für diverse Situationen automatisch dazu gestellt. Anhand dieser können individuelle und dynamische Abbruchkriterien formuliert werden.

Schließlich werde ich noch kurz die TSS Prozesse und ein weiteres Programm vorstellen, welches die bisher betrachteten BPDL Prozesse erweitert auf TSS-Prozesse. Dabei sollen nicht nur das besonders interessante Verhalten vom Wechsel des dominanten Merkmals simuliert werden, sondern es wird eine verbesserte Laufzeit durch Interpolation vorgestellt die eine effiziente Simulation trotz sehr großer Zeit und besonders präzise Betrachtung von Aktionsreichen Gebieten anbietet.

2 Modell

Das verwendete Model lehnt sich an das Model aus [1] an. Dieses nutzt die drei grundlegenden Mechanismen von Darwins Evolutionslehre: Vererbung, Variation (Mutationen) und Selektion durch Wettbewerb um eine Menge von Merkmalen für Individuen zu beschreiben. Diese bestimmen die Fähigkeit des Individuums zu überleben und sich fortzupflanzen.
Jedoch wurden für meine Simulation einige kleine Änderungen gewünscht die im Anschluss erläutert werden.

2.1 Grundlagen

Sei X der Raum der Merkmale. Jedes Individuum hat genau ein solches Merkmal $i \in X$. Der Einfachheit halber sei X eine Indexmenge: $X = \{1, \ldots, n\}$ repräsentativ für eine Durchzählung der Merkmale und im folgenden seien $i, j \in X$ zwei solche Merkmale. Außerdem gilt:

- Jedes Individuum kann sich asexuell fortpflanzen oder sterben.
- Fortpflanzungs- und Todeszeitpunkte können durch sogenannte exponentielle Uhren (wie in [2, S. 3]) beschrieben werden. Diese Uhren haben exponentiell verteilte Weckzeiten. Durch die Gedächtnislosigkeit der Exponentialverteilung, können alle Uhren nach dem ersten Klingeln neu gestellt werden.
- Die Fortpflanzung eines Individuums aus i kann jedoch auch in der Geburt eines Individuums in j resultieren.

Später wird deutlich dass die Zurückstellbarkeit der Uhren entscheiden ist um die Sichtweise von der Ebene des Individuums auf die der gesamten Population zu heben.
Diese Todes und Fortpflanzungs- Ereignisse eines Individuums haben feste Raten die das dazugehörige Merkmal beschreiben.

$b(i)$:	Ist die Geburtenraten durch ein Individuum mit Merkmal i.
$d(i)$:	Ist die natürliche Todesrate.
$c(i, j)$:	Ist die Todesrate durch Wettbewerb zwischen Individuen mit Merkmal i und j.
μ:	Ist die Mutationswahrscheinlichkeit "auf die Nachbarn" mit je $\frac{\mu}{2}$ pro Nachbar.

Schließlich lassen sich durch Superposition z.B. die beiden Todesraten zu einer gemeinsamen Todesrate zusammenfassen oder die arteigene Geburtenrate abtrennen.

$b(i) \cdot (1-\mu)$:	ist die arteigene Geburtenrate eines Individuums mit Merkmal i
$d(i) + \sum_{j=1}^{N_t} c(i,j)$:	ist die gesamte Todesrate eines Individuums mit Merkmal i (mit $N_t :=$ #Individuen zur Zeit t mit Merkmal i).
$d(i) + \sum_{j=1}^{n} c(i,j) \cdot n_t(i)$:	wie oben, nur mit $n :=$ #Merkmale, und $n_t(i) :=$ #Individuen zur Zeit t mit Merkmal i

Die letzte Darstellung der Todesrate ist praktischer für die Betrachtung der Population. Ähnlich können weitere Ereignisse zusammengefasst werden, so dass man z.B. eine Todesrate und eine arteigene Geburtenraten der Merkmale erstellen kann:

- Arteigene Geburtenrate des Merkmals i:

$$b(i) \cdot (1-\mu) + (b(i+1) \cdot \mathbb{1}_{i>1} + b(i-1) \cdot \mathbb{1}_{i>1}) \cdot \frac{\mu}{2}$$

- Gesamte Todesrate des Merkmals i:

$$d(x) + \sum_{i=1}^{n} c(x, x_i) \cdot n_t(x_i)$$

Die Simulation soll die Entwicklung der Merkmale und nicht die Ereignisse der Individuen darstellen, daher ist es unpraktisch diese zu betrachten. Alternativ werden die Ereignisse zu denen von Merkmalen zusammengefasst:

- Geburtenrate (Wachstumsrate) des Merkmals x:

$$B(x) = b(x) \cdot (1-\mu) \cdot n_t(x) + (b(x+1) \cdot n_t(x+1) + b(x-1) \cdot n_t(x-1)) \cdot \frac{\mu}{2}$$

  (Notation deutlich machen)

- Todesrate des Merkmals x:

$$D(x) = d(x) \cdot n_t(x) + n_t(x) \cdot \sum_{i=1}^{n} c(x, x_i) \cdot n_t(x_i)$$

- Die Simulation soll die Entwicklung der Merkmale und nicht die Ereignisse der Individuen darstellen, daher ist es unpraktisch diese zu betrachten. Alternativ werden die Ereignisse zu denen von Merkmalen zusammengefasst:

- Geburtenrate (Wachstumsrate) des Merkmals x:

  $$B(x) = b(x)\cdot(1-\mu)\cdot n_t(x)+(b(x+1)\cdot n_t(x+1)+b(x-1)\cdot n_t(x-1))\cdot\frac{\mu}{2}$$

  (Notation deutlich machen)
- Todesrate des Merkmals x:

  $$D(x) = d(x)\cdot n_t(x) + n_t(x)\cdot\sum_{i=1}^{n} c(x,x_i)\cdot n_t(x_i)$$

  Das entspricht 2 exponentiellen Uhren pro Merkmal. Eine für Tod und eine für Geburt innerhalb des Merkmals.

- Zur praktischen Simulation ist eine Gesamtrate für das Eintreten eines Ereignisses praktischer. Auf diese Weise wird nur auf das eintreffen einer Uhr gewartet.
  - Ereignisrate des Merkmals x (Trait Rate):

    $$TR(x) = B(x) + D(x)$$

  - Totale Ereignis Rate (Total Event Rate):

    $$TER = \sum_{x\in X} TR(x)$$

  Mit der Totalen Ereignisrate gibt es jetzt eine Rate die es erlaubt eine Zufallsvariable für das Eintreffen einer Variable zu ziehen. Anschließend ist es nur noch erforderlich (mit der Ziehung zwei weiterer Zufallsvairablen) festzustellen welchem Merkmal welches Ereignis zukommt.

- Die Population ist ein Markov Sprungprozess der durch Zufallsvariablen

  $$\nu_t = \sum_{i=1}^{N_t}\delta_{x_i},\ \text{mit}\ \int_X 1\nu_t(dx) = N_t$$

  beschrieben wird.
  Dabei gilt:

  $$\nu_t \in M_F(X) = \left\{\sum_{i=1}^{n}\delta_{x_i}, n\in\mathbb{N}, x_1,\ldots,x_n\in X\right\}$$

- (Vielleicht nicht erwähnen?) Der Generator des so definierten BPDL-Prozess ν_t ist:

$$L_{\phi(\nu)} = \int_x b(x)(1-\mu)[\phi(\nu+\delta_x)-\phi(\nu)]\nu(dx) \tag{1}$$
$$+ \int_x \int_{\mathbb{R}^d} b(x)(\mu)[\phi(\nu+\delta_{x+z})-\phi(\nu)]m(x,dz)\nu(dx) \tag{2}$$
$$+ \int_x d(x)[\phi(\nu-\delta_x)-\phi(\nu)]\nu(dx) \tag{3}$$
$$+ \int_x \left(\int_X c(x,y)\nu(dy)\right)[\phi(\nu-\delta_x)-\phi(\nu)]\nu(dx) \tag{4}$$

  mit $\phi : M_F \to \mathbb{R}$

2.2 Normalisierung

2.3 Equilibrium

Man erkennt dass die Merkmale für $K \to \infty$ eine Konvergenz gegen eine Funktion $\xi_t = n_t\delta_x$ aufweist. Diese Funktion wiederum konvergiert gegen einen stabilen Zustand (im folgenden Equilibrium genannt), worin sich die Populationsgröße nicht mehr ändert. Diese sehen für monomorphe und dimorphe Populationen ohne Mutation folgendermaßen aus:

- Monomorphe Population

$$0 = \dot{n} = (b(x) - d(x) - \bar{n}c(x,x))\bar{n}$$
$$\bar{n} = \frac{[b(x)-d(x)]_+}{c(x,x)}$$

- Dimorphe Population

$$n_x = \frac{(b(x)-d(x))c(y,y) - (b(y)-d(y))c(x,y)}{c(y,y)c(x,x) - c(y,x)c(x,y)}$$
$$n = \frac{(b(y)-d(y))c(x,x) - (b(x)-d(x))c(y,x)}{c(y,y)c(x,x) - c(y,x)c(x,y)}$$

  oder $(\bar{n}_x, 0)$, $(0, \bar{n}_y)$ bzw. $(0,0)$

Später im Kapitel TSS wird die Fitness Funktion eingeführt die näher erläutern kann wann welcher dimorphe stabile Zustand angenommen wird. In den Simulationen sind diese stabilen Zustände als gestrichelte Gerade angezeigt, gegen die Konvergenz stattfinden soll.

3 Algorithmus

Der Simulation liegt ein Algorithmus zugrunde der einen Sprung des Markov Sprung Prozesses durchführt. Im Code wird dazu die "EvolutionStep()" Funktion aufgerufen. (Es gibt auch Jump statt Step, aber mehrere Schritte)

Algorithm 1 EvolutionStep()

Ensure: A full evolution Step happened

```
1: calculateEventRates();
2: sampleEventTime();
3: changeATrait();
```

Von dieser werden folgende Berechnungen angestoßen:

Algorithm 2 EvolutionStep()

Ensure: A full evolution Step happened

```
1: —>calculateEventRates();
2: calculateTotalDeathRates()
3: calculateTotalBirthRates()
4: calculateTotalEventRate()
5: —>sampleEventTime();
6: sampleEventTime();
7: —>changeATrait();
8: choseTraitToChange();
9: choseEventType();
10: executeEventTypeOnTrait();
```

Schließlich der Ablauf der tatsächlichen Berechnung:

Algorithm 3 EvolutionStep()

Ensure: A full evolution Step happened

Require: $t, X = \{0, \ldots, n-1\}$

—>calculateEventRates();
for $x \in X$ **do**
  $D(x) := n_t(x) \cdot \left(d(x) + \sum_{y \in X} c(x,y) \cdot n_t(y)\right)$
  $B(x) := \underbrace{b(x) \cdot (1-\mu) \cdot n_t(x)}_{arteigene}$
  if $x > 0$ **then**
    $B(x) += \underbrace{b(x-1) \cdot n_t(x-1)}_{MutationLinks} \cdot \frac{\mu}{2}$
  end if
  if $x < n-1$ **then**
    $B(x) += \underbrace{b(x+1) \cdot n_t(x+1)}_{MutationRechts} \cdot \frac{\mu}{2}$
  end if
  $TotalTraitRate(x) = B(x) + D(x)$
end for
$TotalEventRate := \sum_{x \in X} TotalTraitRate(x)$
—>sampleEventTime();
sample $Z \sim exp(TotalEventRate)$
$t += Z$
—>choseTraitToChange();
sample $Y \sim U(0, TotalEventRate)$
for $x \in X$ **do**
  if $Y \leq TotalTraitRate(x)$ **then**
    $ChosenTrait := x$
    break
  end if
  $Y -= TotalTraitRate(x)$
end for
—>choseEventType();
sample $Y \sim U(0, TotalTraitRate(ChosenTrait))$
if $Y \leq B(ChosenTrait)$ **then**
  isBirht := true
else
  isBirth := false
end if
—>executeEventTypeOnTrait();
if isBirth **then**
  $n_t(ChosenTrait) += 1$
else
  if $n_t(ChosenTrait) \geq 0$ **then**
    $n_t(ChosenTrait) -= 1$
  end if
end if

4 Simulation

5 Korrektheit der Implementation

6 TSS Prozesse

7 Ausblick

- Weiteres Abbruchkriterium = Zeit : sehr einfach zu implementieren.

Literatur

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

[2] Nicolas Fournier and Sylvie Méléard. A microscopic probabilistic description of a locally regulated population and macroscopic approximations. *Ann. Appl. Probab.*, 14(4):1880–1919, 2004.