Ekološko modeliranje i predviđanje

Sveobuhvatni koncept sveučilišnog udžbenika

Branimir Hackenberger

2025

Sadržaj

I Osnove i matematički okvir **1**

1 Uvod u ekološko modeliranje **2**
1.1 Što je ekološko modeliranje? . . . 2
1.2 Definicije, opseg i uloga . . . 3
1.3 Povijesni pregled . . . 4

2 Matematički i statistički temelji **8**
2.1 Linearna algebra i matrični modeli . . . 8
2.2 Diferencijalne i diferentne jednadžbe . . . 12
2.3 Vjerojatnost, statistika i stohastičnost . . . 18
2.3.1 Uvod u stohastičnost u ekološkim sustavima . . . 18
2.3.2 Osnovni koncepti vjerojatnosti . . . 18
2.3.3 Demografska stohastičnost . . . 19
2.3.4 Okolišna stohastičnost . . . 20
2.3.5 Katastrofična stohastičnost . . . 21
2.3.6 Maksimalna vjerojatnost (Maximum Likelihood) . . . 21
2.3.7 Bayesovska statistika . . . 22
2.3.8 Primjeri u ekološkom modeliranju . . . 23
2.3.9 Model selekcija i usporedba . . . 24
2.3.10 Propagacija nesigurnosti . . . 24
2.3.11 Ključne točke za praktičnu primjenu . . . 25
2.4 Računalni alati . . . 25
2.4.1 Uvod u računalne alate za ekološko modeliranje . . . 25
2.4.2 R programski jezik i okruženje . . . 25
2.4.3 Python programski jezik . . . 28
2.4.4 MATLAB . . . 31
2.4.5 GNU Octave . . . 31
2.4.6 Python Scientific Computing Stack . . . 37
2.4.7 Julia programski jezik . . . 44
2.4.8 NetLogo . . . 55
2.4.9 Reproducibilnost i najbolje prakse . . . 57
2.4.10 Integracijska prilazi i interoperabilnost . . . 62
2.4.11 Zaključak i preporuke . . . 62

II Populacije, zajednice i ekosustavi **66**

3 Modeli populacijske dinamike **67**
3.1 Jednostavni modeli . . . 67
3.1.1 Uvod u populacijsku dinamiku . . . 67

3.1.2 Eksponencijalni rast . . . 67
3.1.3 Logistički rast . . . 68
3.1.4 Diskretni modeli rasta . . . 69
3.1.5 Usporedba diskretnih modela . . . 70
3.1.6 Procjena parametara . . . 70
3.2 Dobno/stanovno strukturirani modeli . . . 71
3.2.1 Uvod u strukturirane modele . . . 71
3.2.2 Leslie matrica . . . 71
3.2.3 Lefkovitch matrica . . . 72
3.2.4 Analiza osjetljivosti i elastičnosti . . . 73
3.2.5 Napredni koncepti strukturiranih modela . . . 73
3.3 Metapopulacijski modeli . . . 74
3.3.1 Uvod u metapopulacijsku ekologiju . . . 74
3.3.2 Levinsov klasični model . . . 74
3.3.3 Prostorno eksplicitni metapopulacijski modeli . . . 75
3.3.4 Modeli s heterogenošću fragmenata . . . 76
3.3.5 Diskretni metapopulacijski modeli . . . 76
3.3.6 Fragmentacija staništa . . . 77
3.3.7 Empirijski primjer: Metapopulacija leptira . . . 77
3.3.8 Praktične primjene metapopulacijskih modela . . . 78

4 Međuspecijske interakcije i modeli zajednica 79
4.1 Grabljivac–plijen i funkcijski odgovori . . . 79
4.1.1 Uvod u predator-prey dinamiku . . . 79
4.1.2 Lotka-Volterra modeli . . . 79
4.1.3 Funkcijski odgovori . . . 81
4.1.4 Modifikacije Lotka-Volterra modela . . . 82
4.1.5 Prostorni predator-prey modeli . . . 83
4.2 Konkurencija i koegzistencija . . . 83
4.2.1 Teorija konkurencije . . . 83
4.2.2 Lotka-Volterra model konkurencije . . . 84
4.2.3 Moderna teorija koegzistencije . . . 85
4.2.4 Resource partitioning i niche teorija . . . 86
4.3 Ekološke mreže i stabilnost zajednica . . . 87
4.3.1 Uvod u ekološke mreže . . . 87
4.3.2 Topologija hranidbenih mreža . . . 87
4.3.3 May-ova analiza stabilnosti . . . 88
4.3.4 Strukturni aspekti stabilnosti . . . 88
4.3.5 Kaskadni učinci . . . 89
4.3.6 Dinamička analiza mreža . . . 89
4.3.7 Empirijski primjeri . . . 90
4.3.8 Mrežne metrike i bioraznolikost . . . 90
4.3.9 Prostorno-eksplicitni mrežni modeli . . . 91

5 Modeli ekosustava i bioenergetika 92
5.1 Kruženje tvari i energije . . . 92
5.1.1 Osnovni principi i bilance tvari . . . 92
5.1.2 Biogeokemijski ciklusi . . . 92
5.1.3 Energetski tokovi kroz ekosustav . . . 93
5.1.4 Ohmov zakon za ekologiju . . . 94
5.2 Bioenergetski i DEB modeli . . . 94

5.2.1 Osnove DEB teorije . . . 94
5.2.2 Standardni DEB model . . . 95
5.2.3 Funkcionalni odgovor u DEB kontekstu . . . 96
5.2.4 Povezivanje DEB modela s populacijskim dinamikama . . . 96
5.3 Ecosystem services . . . 96
5.3.1 Klasifikacija i kvantifikacija . . . 96
5.3.2 Ekonomska vrednovanje . . . 97
5.3.3 Trade-off analize . . . 97
5.3.4 Prostorno modeliranje usluga . . . 98
5.3.5 Dinamičko modeliranje usluga . . . 98

III Prostor, predviđanje i AI 100

6 Prostorno-ekološko modeliranje 101
6.1 GIS, daljinska istraživanja i prostorne skale . . . 101
6.1.1 Geografski informacijski sustavi (GIS) u ekologiji . . . 101
6.1.2 Daljinska istraživanja i satelitski podatci . . . 101
6.1.3 Prostorne skale u ekologiji . . . 102
6.1.4 Prostorna autokorelacija . . . 103
6.2 Modeli rasprostiranja vrsta (SDM/ENM) . . . 103
6.2.1 Konceptualni okvir . . . 104
6.2.2 Klimatske varijable i bioklimatski slojevi . . . 104
6.2.3 Algoritmi SDM modeliranja . . . 105
6.2.4 Evaluacija modela . . . 106
6.2.5 Pristranost uzorkovanja . . . 106
6.3 Individualno temeljeni modeli (IBM) . . . 107
6.3.1 Konceptualni okvir IBM-a . . . 107
6.3.2 Prostorno ponašanje i movement . . . 107
6.3.3 Energetski modeli . . . 108
6.3.4 Reprodukcija i demografija . . . 108
6.3.5 Interakcije između jedinki . . . 108
6.3.6 Emergentna svojstva . . . 109
6.3.7 Implementacija i softverski alati . . . 109

7 Prediktivno modeliranje i scenariji 111
7.1 Scenariji klimatskih promjena . . . 111
7.2 Predviđanje invazija i rani sustavi upozorenja . . . 111
7.3 Spajanje mehanističkih i statističkih pristupa . . . 111

8 Statistički i AI pristupi 112
8.1 Klasična i Bayesovska statistika . . . 112
8.2 Strojno učenje . . . 112
8.3 Duboko učenje i digitalni blizanci ekosustava . . . 112

IV Validacija, nesigurnost i primjene 113

9 Validacija, verifikacija i osjetljivost 114
9.1 Kalibracija i verifikacija . . . 114
9.2 Analize osjetljivosti i robustnosti . . . 114

9.3 Etika i komunikacija nesigurnosti 114

10 Primjene u upravljanju okolišem **115**
10.1 Zaštita prirode i bioraznolikosti 115
10.2 Poljoprivreda i šumarstvo 115
10.3 Ekotoksikologija i procjena rizika 115
10.4 Klimatske politike i održivi razvoj 115

V Studije slučaja i praktične vježbe **116**

11 Studije slučaja **117**
11.1 Gujavice u agroekosustavima 117
11.2 Širenje komaraca 117
11.3 Eutrofikacija riječnog ekosustava 117
11.4 Fragmentacija šuma i ptice metapopulacija 117
11.5 Bayesove mreže u ekotoksikologiji 117

12 Praktične vježbe (R i Python) **118**
12.1 Uvod u R i Python alate 118
12.2 Leslie matrica u R 118
12.3 Lotka–Volterra simulacija u Python 118
12.4 SDM u R (skica s `dismo`) 119
12.5 Random Forest za invazije 119
12.6 DEB simulacije (skica) 119

A Matematički prilozi **121**
A.1 Osnovni populacijski modeli 121

B Instalacija softvera **122**

C Primjeri koda **123**

D Rječnik pojmova **124**

Predgovor

Ovaj udžbenik sintetizira teorijske temelje, metode i praktične primjene ekološkog modeliranja s posebnim naglaskom na validaciju, nesigurnost i prediktivne scenarije. Struktura je objedinjena iz dvije početne skice i prilagođena kao koherentan kurikulum za preddiplomske i diplomske studije ekologije, biologije i znanosti o okolišu, te kao referentni priručnik za istraživače i praktičare.[1]

[1]Temeljni raspored poglavlja, definicije i okviri preuzeti su i integrirani iz *Koncept.pdf* i *Koncept2.pdf*.

Dio I

Osnove i matematički okvir

Poglavlje 1

Uvod u ekološko modeliranje

1.1 Što je ekološko modeliranje?

Ekološko modeliranje predstavlja interdisciplinarno područje koje koristi matematičke, statističke i računalne alate za opisivanje, razumijevanje i predviđanje ekoloških procesa i obrazaca. Ova definicija, premda koncizna, skriva složenost i širinu discipline koja je postala nezaobilazan dio moderne ekološke znanosti.

U svojoj srži, ekološko modeliranje nastoji prevesti složene prirodne procese u formalne, kvantitativne okvire koji omogućavaju sistemsku analizu i razumijevanje. Ova transformacija složenih ekoloških sustava u matematičke reprezentacije nije samo akademska vježba, već predstavlja moćan alat za rješavanje praktičnih problema u zaštiti okoliša, upravljanju prirodnim resursima i predviđanju ekoloških promjena.

Važnost ekološkog modeliranja proizlazi iz fundamentalne prirode ekoloških sustava. Prirodni ekosustavi karakteriziraju nelinearne interakcije između brojnih komponenti, višestruke povratne veze, prostorna i vremenska varijabilnost, te inherentna nesigurnost. Tradicionalni eksperimentalni pristup, premda neprocjenjiv, često je ograničen u svojoj sposobnosti obuhvaćanja pune složenosti ekoloških procesa. Modeliranje pruža okvir za integraciju znanja iz različitih izvora, omogućava testiranje hipoteza na načine koji nisu praktični ili etični u prirodi, te pomaže u razumijevanju emergentnih svojstava koja proizlaze iz složenih interakcija.

Ekološki modeli služe kao konceptualni mostovi između teorije i empirijskih opažanja. Oni omogućavaju istraživačima da formaliziraju svoje razumijevanje ekoloških procesa, testiraju konzistentnost svojih hipoteza i identificiraju ključne znanja koja nedostaju. Kroz proces modeliranja, često otkrivamo da naše intuitivno razumijevanje složenih sustava može biti nepotpuno ili čak pogrešno, što pokreće daljnja istraživanja i poboljšava naše znanje.

Primjene ekološkog modeliranja protežu se kroz spektar prostornih i vremenskih skala, od molekularnih interakcija unutar organizama do globalnih biogeokemijskih ciklusa koji se odvijaju kroz tisućljeća. Na individualnoj razini, modeli mogu opisivati fiziološke procese poput energetskog metabolizma ili reproduktivnih ciklusa. Na populacijskoj razini, modeliranje pomaže u razumijevanju dinamike brojnosti, strukture starosti i prostornog rasprostiranja vrsta. Modeli zajednica i ekosustava fokusiraju se na međuspecijske interakcije, tokove energije i hranjive tvari, te održavanje bioraznolikosti.

Jedan od ključnih aspekata ekološkog modeliranja je njegova sposobnost integracije znanja iz različitih disciplina. Moderna ekološka istraživanja sve više postaju interdisciplinarna, kombinirajući principe iz biologije, kemije, fizike, geografije, matematike i informatike. Modeliranje omogućava sinteze znanja iz ovih različitih područja u koherentne okvire koji mogu pružiti nova uvida u funkcioniranje prirodnih sustava.

Tehnološki napredak, posebice u područjima računalnih znanosti i umjetne inteligencije, značajno je proširio mogućnosti ekološkog modeliranja. Dostupnost velikih baza podataka,

poboljšani algoritmi strojnog učenja i povećana računalna moć omogućili su razvoj sofisticiranih modela koji mogu obraditi kompleksne skupove podataka i uhvatiti suptilne obrasce koji prije nisu bili uočljivi.

Međutim, ekološko modeliranje nije bez izazova. Prirodna varijabilnost, nelinearnost sustava, ograničenja podataka i inherentne nesigurnosti predstavljaju stalne izazove modelarima. Uspješno modeliranje zahtijeva ne samo tehničku ekspertizu, već i duboko razumijevanje ekoloških principa, kritičko mišljenje o ograničenjima modela i sposobnost komunikacije rezultata različitim auditorijima.

1.2 Definicije, opseg i uloga

Ekološko modeliranje može se klasificirati prema svojim primarnim ciljevima i funkcijama. Ova klasifikacija pomaže u razumijevanju različitih pristupa modeliranju i njihovih specifičnih primjena u ekološkoj znanosti i upravljanju okolišem.

Deskriptivno modeliranje predstavlja temeljnu razinu ekološkog modeliranja čiji je primarni cilj opisivanje i kvantifikacija opaženih obrazaca u prirodi. Ovi modeli nastoje odgovoriti na pitanje "što se događa?" kroz sistemsku analizu i sažimanje empirijskih podataka. Deskriptivni modeli često služe kao prvi korak u razumijevanju složenih ekoloških fenomena, omogućavajući identificiranje ključnih varijabli, otkrivanje obrazaca i trendova te karakterizaciju varijabilnosti sustava.

Primjeri deskriptivnog modeliranja uključuju analizu vremenskih serija populacijskih abundance, karakterizaciju prostornih obrazaca rasprostiranja vrsta, kvantifikaciju sezonskih promjena u produktivnosti ekosustava ili opisivanje strukture ekoloških mreža. Regresijski modeli koji opisuju odnose između abundancije vrsta i čimbenika okoliša također spadaju u ovu kategoriju. Premda mogu izgledati jednostavno, deskriptivni modeli često zahtijevaju sofisticirane statističke tehnike, posebice kada se suočavaju s kompleksnim, višedimenzionalnim skupovima podataka.

Važnost deskriptivnog modeliranja ne smije se podcjenjivati. Precizno opisivanje prirodnih obrazaca predstavlja temelj za sve daljnje analize i interpretacije. Bez dobrog deskriptivnog razumijevanja, pokušaji objašnjavanja ili predviđanja mogu biti usmjereni pogrešno ili temeljeni na netočnim pretpostavkama.

Eksplanatorno modeliranje ide korak dalje od pukog opisivanja te nastoji objasniti zašto se opaženi obrasci pojavljuju. Ovi modeli fokusiraju se na identificiranje i kvantifikaciju uzročno-posljedičnih veza između različitih komponenti ekoloških sustava. Eksplanatorno modeliranje odgovara na pitanje "zašto se to događa?" kroz testiranje hipoteza o mehanizmima koji pokreću opažene procese.

Mechanistički pristup eksplanatornog modeliranja temelji se na razumijevanju osnovnih biologijskih, kemijskih i fizičkih procesa koji djeluju u ekološkim sustavima. Na primjer, modeli populacijske dinamike koji uključuju specifične parametre rođenja, smrti, imigracije i emigracije nastoje objasniti promjene u brojnosti populacije kroz identificiranje ključnih demografskih procesa. Slično tome, modeli međuspecijskih interakcija pokušavaju objasniti koegzistenciju ili kompetitivno isključivanje kroz kvantifikaciju specifičnih mehanizama poput kompeticije za resurse ili grabljivačkih odnosa.

Eksplanatorno modeliranje često uključuje usporedbu alternativnih hipoteza ili mehanizama kroz formalne tehnike poput usporedbe modela ili Bayesovske inferencije. Ovaj pristup omogućava ne samo identificiranje najvjerojatnijih objašnjenja, već i kvantifikaciju nesigurnosti oko različitih hipoteza.

Prediktivno modeliranje usmjereno je na predviđanje budućih stanja ili ponašanja ekoloških sustava. Ova kategorija modeliranja odgovara na pitanje "što će se dogoditi?" i predstavlja kritičnu komponentu mnogih aplikacija u upravljanju okolišem i politici zaštite

prirode. Prediktivni modeli mogu se fokusirati na kratkoročna predviđanja (npr. sezonske promjene u abundanciji vrsta) ili dugoročne projekcije (npr. utjecaji klimatskih promjena na bioraznolikost).

Uspješno prediktivno modeliranje zahtijeva kombinaciju dobrog razumijevanja sustava (često proizašlog iz deskriptivnih i eksplanatornih analiza) i robusnih statističkih ili matematičkih tehnika. Modeli rasprostiranja vrsta koji predviđaju buduća staništa pod različitim klimatskim scenarijima predstavljaju primjer prediktivnog modeliranja. Slično tome, populacijski modeli koji projektiraju buduće veličine populacija pod različitim upravljačkim scenarijima spadaju u ovu kategoriju.

Važan aspekt prediktivnog modeliranja je procjena i komunikacija nesigurnosti. Prirodni sustavi su inherentno varijabilni i složeni, što znači da predviđanja uvijek nose određenu razinu nesigurnosti. Kvalitetno prediktivno modeliranje mora ovu nesigurnost kvantificirati i prenijeti je krajnjim korisnicima na razumljiv način.

Preskriptivno modeliranje predstavlja najnapredniju razinu ekološkog modeliranja te nastoji pružiti preporuke za upravljanje ili intervencije. Ovi modeli odgovaraju na pitanje "što trebamo učiniti?" kombinirajući znanje o tome kako sustavi funkcioniraju s ciljevima upravljanja ili očuvanja. Preskriptivni modeli često integriraju ekološke, ekonomske i socijalne faktore te nastoje identificirati optimalne strategije upravljanja.

Optimizacijski pristupi u preskriptivnom modeliranju mogu uključivati linearno ili nelinearno programiranje, dinamičko programiranje ili metaheurističke algoritme. Na primjer, modeli za planiranje zaštićenih područja nastoje identificirati kombinacije lokacija koje maksimiziraju očuvanje bioraznolikosti uz minimiziranje troškova ili konflikata s drugim korisnicima zemljišta. Slično tome, modeli upravljanja ribljim zajednicama mogu preporučiti kvote izlova koje maksimiziraju dugoročnu održivost uz ekonomsku isplativost.

Preskriptivno modeliranje često zahtijeva uključivanje različitih interesnih skupina u proces definiranja ciljeva i ograničenja. Ovi modeli moraju balansirati različite, često konfliktne ciljeve te uzeti u obzir praktična ograničenja implementacije.

Važno je napomenuti da ove četiri kategorije nisu međusobno isključive. U praksi, kompleksni projekti modeliranja često kombiniraju elemente iz svih četirih pristupa. Deskriptivna analiza može identificirati ključne obrasce, eksplanatorno modeliranje može objasniti uzroke tih obrazaca, prediktivno modeliranje može projektirati buduće scenarije, a preskriptivno modeliranje može preporučiti odgovarajuće upravljačke akcije.

Kombinacija različitih pristupa modeliranja stvara sinergije koje omogućavaju dublje razumijevanje i efikasniju primjenu ekoloških znanja. Ova integracija predstavlja suštinu moderne primijenjene ekologije te ilustrira kako matematički i računalni alati mogu služiti kao mostovi između teorijskog razumijevanja i praktičnih rješenja za očuvanje i upravljanje prirodnim sustavima.

1.3 Povijesni pregled

Razvoj ekološkog modeliranja može se pratiti kroz evoluciju matematičkih pristupa opisivanju prirodnih procesa, pri čemu svaki značajan doprinos gradi na prethodnim spoznajama i odgovara na nova pitanja koja se pojavljuju s napretkom znanosti. Ovaj povijesni pregled prati ključne prekretnice koje su oblikovale modernu disciplinu ekološkog modeliranja, od ranih demografskih teorija do sofisticiranih matričnih pristupa koji i danas čine temelj populacijske ekologije.

Malthusovi temelji demografskog modeliranja

Počeci formalnog pristupa modeliranju populacijskih procesa mogu se pratiti do rada Thomasa Roberta Malthusa (1766–1834), engleskog ekonomista i demografa čiji je utjecaj daleko nadišao granice ekonomske znanosti. U svom revolucionarnom djelu "An Essay on the

Principle of Population" (1798), Malthus je postavio temelj za kvantitativno razumijevanje populacijske dinamike kroz formulaciju jednostavnog, ali moćnog koncepta eksponencijalnog rasta.

Malthusova osnovna pretpostavka bila je da se populacije, u odsutnosti ograničavajućih faktora, povećavaju u geometrijskoj progresiji, dok se resursi potrebni za održavanje tih populacija povećavaju samo u aritmetičkoj progresiji. Ova asimetrija između potencijala rasta populacije i dostupnosti resursa predstavlja temeljni paradoks koji je Malthus identificirao kao uzrok neizbježnih kriza u ljudskim društvima.

Matematički, Malthusov model može se izraziti diferencijalnom jednadžbom:

$$\frac{dN}{dt} = rN$$

gdje je $N(t)$ veličina populacije u vremenu t, a r predstavlja intrinsičnu stopu rasta populacije. Rješenje ove jednadžbe daje eksponencijalnu funkciju $N(t) = N_0 e^{rt}$, gdje je N_0 početna veličina populacije. Premda Malthus nije koristio ovu formalnu matematičku notaciju, njegova konceptualna formulacija bila je ekvivalentna ovom modelu.

Značaj Malthusova doprinosa leži ne samo u prepoznavanju eksponencijalnog karaktera neograničenog rasta, već i u razumijevanju da takav rast nije održiv u konačnom svijetu. Ova spoznaja postavila je temelje za kasnije modele koji su nastojali uključiti ograničavajuće faktore i opisati realnije scenarije populacijske dinamike.

Malthusove ideje imale su dalekosežan utjecaj na razvoj demografije, ekonomije i, što je posebno važno za našu raspravu, na Charles Darwinovo razumijevanje prirodne selekcije. Darwin je eksplicitno priznao da ga je čitanje Malthusova eseja inspiriralo na formulaciju principa opstanka najsposobnijih, što ilustrira povezanost između kvantitativnih modela i temeljnih biologijskih teorija.

Verhulstova inovacija: uvođenje nosivosti staništa

Pierre-François Verhulst (1804–1849), belgijski matematičar i demograf, prepoznao je fundamentalno ograničenje Malthusova modela: u stvarnosti, nijedna populacija ne može rasti eksponencijalno neograničeno. Verhulstov ključni uvid bio je da se stopa rasta populacije mora smanjivati kako se populacija približava granicama koje joj nameću dostupni resursi ili drugi čimbenici okoliša.

Godine 1838. Verhulst je uveo ono što je nazvao "logističkom jednadžbom":

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$$

gdje je K nova konstanta koju je Verhulst nazvao "nosivost staništa" (carrying capacity). Ovaj termin opisuje maksimalnu veličinu populacije koju određeni okoliš može održati neograničeno, uzimajući u obzir dostupnost hranjivih tvari, prostora, i drugih resursa.

Elegantnost Verhulstova modela leži u njegovoj jednostavnosti i intuitivnosti. Kada je populacija mala ($N \ll K$), izraz u zagradi približava se jedinici, pa se model ponaša kao Malthusov eksponencijalni model. Međutim, kako populacija raste i približava se nosivosti staništa, izraz $(1 - N/K)$ postaje sve manji, što rezultira smanjenjem efektivne stope rasta. Kada populacija dosegne nosivost ($N = K$), stopa rasta postaje nula, što znači da je populacija u ravnoteži.

Rješenje logističke jednadžbe daje sigmoidu:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}}$$

Ova krivulja karakterizira početnu fazu sporeg rasta, zatim fazu brzog eksponencijalnog rasta, i konačno fazu usporavanja kako se populacija približava asimptoti definiranoj nosivosti staništa.

Verhulstov model predstavljao je paradigmatski pomak u razumijevanju populacijske dinamike. Prvi put je formalno uključio koncept ograničavajućih faktora i predstavio ideju da sama veličina populacije može djelovati kao regulacijski mehanizam kroz intraspecijsku konkurenciju. Ovaj model postavio je temelje za razumijevanje density-dependent regulacije populacija, koncepta koji i danas predstavlja središnji element populacijske ekologije.

Lotka-Volterrini modeli: rođenje međuspecijskog modeliranja

Razvoj modela koji opisuju interakcije između različitih vrsta označava sljedeću važnu fazu u evoluciji ekološkog modeliranja. Alfred James Lotka (1880–1949) i Vito Volterra (1860–1940), radivši neovisno jedan o drugome u ranim desetljećima 20. stoljeća, razvili su sustave diferencijalnih jednadžbi koji opisuju dinamiku dvije vrste u interakciji.

Volterra, talijanski matematičar i fizičar, potaknut je na ovaj rad praktičnim problemom. Njegov kolega, italijanski biolog Umberto D'Ancona, primijetio je neočekivane promjene u ulovoj ribljih zajednica u Jadranskom moru tijekom Prvog svjetskog rata. Konkretno, udio grabljivačkih riba u ukupnom ulovu značajno se povećao tijekom rata, kada je ribolov bio ograničen. D'Ancona je zatražio Volterrin savjet za objašnjenje ovog fenomena.

Volterra je pristup ovom problemu kroz matematičku analizu grabljivac-plijen sustava, razvivši sustav jednadžbi koji opisuje dinamiku dvije vrste:

$$\frac{dN}{dt} = rN - aNP$$
$$\frac{dP}{dt} = eaNP - mP$$

gdje N predstavlja abundanciju plijena, P abundanciju grabljivca, r je intrinsična stopa rasta plijena, a je stopa napada grabljivca, e je efikasnost konverzije plijena u potomstvo grabljivca, a m je stopa smrti grabljivca.

Istovremeno, Lotka je razvijao slične modele u kontekstu kemijskih reakcija i općenite teorije sustava, pri čemu je prepoznao da se isti matematički formalizam može primijeniti na različite prirodne procese, uključujući međuspecijske interakcije u ekologiji.

Analiza Lotka-Volterrinih jednadžbi otkriva fascinantna svojstva. Sustav ima neutralnu stabilnost, što znači da oscilira u zatvorenim orbitama oko ravnotežne točke. Ove oscilacije imaju konstantnu amplitudu i period koji ovisi o početnim uvjetima, ali ne konvergiraju k ravnotežnoj točki niti se udaljavaju od nje. Ovakvo ponašanje predstavlja idealizirani scenarij u kojem grabljivac i plijen osciliraju u savršenoj sinkronizaciji.

Značaj Lotka-Volterrinih modela nadilazi njihovu specifičnu primjenu na grabljivac-plijen sustave. Oni predstavljaju prvi formalni pristup modeliranju međuspecijskih interakcija i postavljaju temelje za razumijevanje dinamike složenih ekoloških zajednica. Dodatno, uvode važne koncepte poput funkcijskih odgovora (kako se stopa predacije mijenja s abundancijom plijena) i vremenskih kašnjenja u ekološkim procesima.

Premda osnovni Lotka-Volterrin model čini značajne pretpostavke (poput nepostojanja nosivosti staništa za plijen i konstantnih parametara), on je stimulirao razvoj brojnih proširenja i modifikacija. Ovi uključuju uvođenje nosivosti staništa za plijen, alternativnih funkcijskih odgovora, prostorne heterogenosti, i vremenskih varijacija u parametrima.

Lesliejeve matrice: revolucija u strukturiranom modeliranju

Patrick Holt Leslie (1900–1972), škotski biomatememačar, revolucionizirao je populacijsko modeliranje uvođenjem matričnih pristupa koji omogućavaju opis populacija s kompleksnom dobnom ili stadijskom strukturom. Leslie je prepoznao da je jednostavno tretiranje populacije kao homogene cjeline često nerealno, jer se demografski parametri (stope rođenja, smrti, reprodukcije) značajno razlikuju između različitih dobnih skupina.

Leslie matrica, formulirana 1940-ih godina, predstavlja način opisivanja populacijske dinamike kroz diskretne vremenske korake, pri čemu se populacija dijeli u dobne klase ili razvojne stadije. Osnovna forma Leslie matrice je:

$$\mathbf{L} = \begin{pmatrix} F_1 & F_2 & F_3 & \cdots & F_k \\ P_1 & 0 & 0 & \cdots & 0 \\ 0 & P_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & P_{k-1} & 0 \end{pmatrix}$$

gdje F_i predstavlja plodnost i-te dobne klase (broj potomaka koji će preživjeti do prve dobne klase), a P_i predstavlja vjerojatnost preživljavanja iz i-te u $(i+1)$-vu dobnu klasu.

Populacijski vektor u vremenu $t+1$ može se izračunati kao:

$$\mathbf{n}(t+1) = \mathbf{L}\mathbf{n}(t)$$

gdje $\mathbf{n}(t) = [n_1(t), n_2(t), \ldots, n_k(t)]^T$ predstavlja vektor abundancija u različitim dobnim klasama.

Moć Lesliejevih matrica leži u njihovoj sposobnosti uhvaćanja ključnih aspekata populacijske strukture i dinamike. Dominantna vlastita vrijednost matrice (λ_1) predstavlja asimptotsku stopu rasta populacije, dok odgovarajući vlastiti vektor opisuje stabilnu dobnu strukturu ka kojoj populacija konvergira. Ako je $\lambda_1 > 1$, populacija raste; ako je $\lambda_1 < 1$, populacija opada; a ako je $\lambda_1 = 1$, populacija je u ravnoteži.

Leslie je također razvio koncepte osjetljivosti i elastičnosti, koji omogućavaju kvantifikaciju utjecaja promjena u demografskim parametrima na populacijski rast. Osjetljivost mjeri apsolutnu promjenu u λ s obzirom na malu promjenu u elementu matrice, dok elastičnost mjeri proporcionalnu promjenu. Ovi koncepti postali su fundamentalni alati u populacijskoj biologiji i biologiji zaštite prirode.

Značaj Lesliejevih matrica transcendira njihovu izvornu primjenu. One predstavljaju mostove između individualnih demografskih procesa i populacijskih obrazaca, omogućavaju rigoroznu analizu demografskih podataka, i pružaju okvir za razumijevanje evolucijskih strategija životnog ciklusa. Dodatno, matični pristup omogućava lako proširivanje na složenije strukture, poput metapopulacija ili zajednica s međuspecijskim interakcijama.

Sinteza i naslijeđe

Ovi pionirski doprinosi uspostavili su temeljna načela ekološkog modeliranja koja i danas oblikuju disciplinu. Malthusov eksponencijalni model uspostavio je kvantitativni pristup demografiji i prepoznao fundamentalni potencijal rasta organizama. Verhulstov logistički model uveo je koncepte ograničavajućih faktora i density-dependent regulacije. Lotka-Volterrine jednadžbe otvorile su područje međuspecijskog modeliranja i pokazale kako se složeni ekološki obrasci mogu nastati iz jednostavnih interakcija. Lesliejeve matrice omogućile su rigorozno tretiranje struktuiranih populacija i povezale individualnu demografiju s populacijskim obrascima.

Svaki od ovih doprinosa predstavlja ne samo tehnički napredak, već i konceptualnu inovaciju koja je proširila naše razumijevanje prirodnih procesa. Oni ilustriraju evolucijski karakter znanosti, gdje se nova znanja grade na prethodnim temeljima, istovremeno proširujući granice mogućeg. Kombinacija matematičke elegantnosti, biološke relevantnosti i praktične primjenjivosti ovih modela osigurava im trajno mjesto u kanonu ekološke znanosti.

Moderna ekološka modeliranja nastavlja se oslanjati na ove temelje, proširujući ih kroz inkorporaciju stohastičnosti, prostorne heterogenosti, klimatskih promjena, i sve sofisticiranijih statističkih i računalnih tehnika. Međutim, osnovna načela identificirana od strane ovih pionira – važnost kvantifikacije, potreba za uključivanjem ograničavajućih faktora, značaj međuspecijskih interakcija, i vrijednost strukturiranog pristupa – ostaju središnji elementi uspješnog ekološkog modeliranja.

Poglavlje 2

Matematički i statistički temelji

2.1 Linearna algebra i matrični modeli

Matrični pristup populacijskom modeliranju predstavlja jedan od najelegantnijih i najmoćnijih alata u kvantitativnoj ekologiji. Ovaj pristup omogućava rigoroznu analizu populacija s kompleksnom demografskom strukturom, pružajući duboke uvide u populacijsku dinamiku kroz primjenu koncepata linearne algebre. Leslie i Lefkovitch matrice, kao temeljni predstavnici ovog pristupa, revolucionizirali su način na koji razumijemo i predviđamo populacijske promjene u strukturiranim populacijama.

Osnove matričnog populacijskog modeliranja

Matrično populacijsko modeliranje temelji se na fundamentalnoj ideji da se kompleksne populacije mogu podijeliti u diskretne klase na temelju dobi, veličine, razvojnog stadija ili drugih demografski relevantnih karakteristika. Umjesto tretiranja populacije kao homogene cjeline, ovaj pristup prepoznaje da se demografski parametri (stope rođenja, smrti, reprodukcije, tranzicije između stadija) značajno razlikuju između različitih skupina unutar populacije.

Osnovna forma matričnog modela populacijske dinamike može se izraziti kao:

$$\mathbf{n}(t+1) = \mathbf{A}\mathbf{n}(t)$$

gdje je $\mathbf{n}(t) = [n_1(t), n_2(t), \ldots, n_k(t)]^T$ vektor koji opisuje broj individua u svakoj od k klasa u vremenu t, a $\mathbf{A}$ je $k \times k$ projekcijska matrica koja sadrži sve demografske parametre koji usmjeravaju tranzicije između klasa i produkciju novih individua.

Iterativnom primjenom ove jednadžbe možemo pratiti evoluciju populacijske strukture kroz vrijeme:

$$\mathbf{n}(t) = \mathbf{A}^t\mathbf{n}(0)$$

Ovaj deceptivno jednostavan izraz skriva bogato matematičko ponašanje koje omogućava duboke uvide u populacijsku dinamiku.

Leslie matrice: klasifikacija po dobi

Leslie matrica, nazvana prema škotskom biomatememematičaru Patricku Holtu Leslieju, predstavlja specijaliziranu formu projekcijske matrice namijenjena modeliranju populacija klasificiranih po dobi. Ova matrica ima specifičnu strukturu koja odražava biološke realnosti demografskih procesa kod organizma s diskretnim reproduktivnim sezonama.

Općenita forma Leslie matrice je:

$$\mathbf{L} = \begin{pmatrix} F_1 & F_2 & F_3 & \cdots & F_{k-1} & F_k \\ P_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & P_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & P_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & P_{k-1} & 0 \end{pmatrix}$$

Elementi prvog reda, F_i, predstavljaju parametre plodnosti (fecundity) za i-tu dobnu klasu. Preciznije, F_i označava broj ženskog potomstva koji će individua iz i-te dobne klase proizvesti i koji će preživjeti do prve dobne klase u sljedećem vremenskom koraku. Ovi parametri su produkt nekoliko komponenti:

$$F_i = m_i \cdot s_0 \cdot \frac{1}{2}$$

gdje je m_i broj potomaka koje rađa individua iz i-te dobne klase, s_0 je vjerojatnost preživljavanja potomstva do prve dobne klase, a faktor $\frac{1}{2}$ pretpostavlja ravnotežu spolova (ako modeliramo samo ženke).

Elementi subdiagonale, P_i, predstavljaju vjerojatnosti preživljavanja (survival probabilities) iz i-te u $(i+1)$-vu dobnu klasu. Formalno:

$$P_i = \mathbb{P}(\text{individua preživljava iz klase } i \text{ u klasu } i+1)$$

Nulti elementi u ostalim pozicijama matrice odražavaju biološke pretpostavke Leslie modela: individue se mogu reproducirati ali ne mogu "ići unazad" u mlađe dobne klase, niti mogu "preskočiti" dobne klase.

Svojstva Leslie matrica i spektralna analiza

Ključ razumijevanja ponašanja Leslie matrica leži u njihovoj spektralnoj analizi - studiju vlastite vrijednosti i vlastite vektora. Za Leslie matricu $\mathbf{L}$, vlastite vrijednosti λ su rješenja karakteristične jednadžbe:

$$\det(\mathbf{L} - \lambda \mathbf{I}) = 0$$

što daje karakteristični polinom:

$$\lambda^k - F_1\lambda^{k-1} - F_2P_1\lambda^{k-2} - F_3P_1P_2\lambda^{k-3} - \cdots - F_kP_1P_2\cdots P_{k-1} = 0$$

Prema Perron-Frobeniusovom teoremu, za Leslie matrice koje zadovoljavaju određene biološke uvjete (primitivity - što je obično slučaj kad bar jedna mlada dobna klasa može se reproducirati i kad postoji pozitivna vjerojatnost preživljavanja između uzastopnih dobnih klasa), postoji jedinstvena dominantna vlastita vrijednost λ_1 koja je:

1. **Realna i pozitivna**: $\lambda_1 > 0$
2. **Jednostruka**: algebarska i geometrijska kratnost je 1
3. **Dominantna**: $|\lambda_i| < \lambda_1$ za sve $i \neq 1$
4. **Ima pozitivni vlastiti vektor**: $\mathbf{w}_1 > 0$

Dominantna vlastita vrijednost i asimptotska stopa rasta

Dominantna vlastita vrijednost λ_1 ima fundamentalnu biološku interpretaciju kao asimptotska stopa rasta populacije. Ova interpretacija proizlazi iz asimptotskog ponašanja iteriranog matričnog modela.

Kada projiciramo populaciju daleko u budućnost, dominantna vlastita vrijednost određuje ponašanje:

$$\lim_{t\to\infty} \frac{|\mathbf{n}(t+1)|}{|\mathbf{n}(t)|} = \lambda_1$$

gdje $|\mathbf{n}(t)|$ označava ukupnu veličinu populacije u vremenu t.

Preciznije, asimptotsko ponašanje populacije može se opisati kao:

$$\mathbf{n}(t) \approx C\lambda_1^t \mathbf{w}_1 \quad \text{za velike } t$$

gdje je C konstanta određena početnim uvjetima, a $\mathbf{w}_1$ je dominantni desni vlastiti vektor (stabilan dobni raspored).

Ova konvergencija ima duboke biološke implikacije:

- **Ako** $\lambda_1 > 1$: populacija raste eksponencijalno stopom $r = \ln(\lambda_1)$ - **Ako** $\lambda_1 < 1$: populacija opada eksponencijalno i konvergira prema izumiranju - **Ako** $\lambda_1 = 1$: populacija je u demografskoj ravnoteži

Stabilna dobna distribucija

Dominantni desni vlastiti vektor $\mathbf{w}_1$ predstavlja stabilnu dobnu distribuciju (stable age distribution) ka kojoj sve populacije s istom Leslie matricom konvergiraju neovisno o početnim uvjetima. Ovaj vektor zadovoljava:

$$\mathbf{L}\mathbf{w}_1 = \lambda_1 \mathbf{w}_1$$

Normaliziranjem ovog vektora tako da $\sum_{i=1}^{k} w_{1i} = 1$, dobivamo proporcionalnu dobnu strukturu:

$$\mathbf{c} = \frac{\mathbf{w}_1}{\sum_{i=1}^{k} w_{1i}}$$

gdje c_i predstavlja proporciju populacije u i-toj dobnoj klasi u stabilnom stanju.

Stabilna dobna distribucija može se eksplicitno izračunati. Za Leslie matricu, elementi stabilne dobne distribucije su:

$$c_1 = \frac{1}{1 + \frac{P_1}{\lambda_1} + \frac{P_1 P_2}{\lambda_1^2} + \cdots + \frac{P_1 P_2 \cdots P_{k-1}}{\lambda_1^{k-1}}}$$

$$c_i = c_1 \cdot \frac{P_1 P_2 \cdots P_{i-1}}{\lambda_1^{i-1}} \quad \text{za } i = 2, 3, \ldots, k$$

Reproduktivna vrijednost

Lijevi vlastiti vektor $\mathbf{v}_1$ odgovarajuće dominantne vlastite vrijednosti ima interpretaciju reproduktivne vrijednosti (reproductive value). Ovaj koncept, uveo ga je R.A. Fisher, opisuje relativni doprinos različitih dobnih klasa budućem reproduktivnom outputu populacije:

$$\mathbf{v}_1^T \mathbf{L} = \lambda_1 \mathbf{v}_1^T$$

Reproduktivna vrijednost i-te dobne klase, v_{1i}, može se interpretirati kao očekivani broj potomaka koje će individua trenutno u i-toj dobnoj klasi proizvesti tijekom ostatka svog života, diskontiran asimptotskom stopom rasta populacije.

Lefkovitch matrice: generalizacija na stanja

Lefkovitch matrice, nazvane prema Leonard P. Lefkovitchu, predstavljaju generalizaciju Leslie matrica na populacije klasificirane po bilo kakvim stanjima, a ne nužno po dobi. Ova generalizacija omogućava modeliranje organizama kod kojih je veličina, razvojni stadij, ili neko drugo obilježje relevantnije za demografiju od kronološke dobi.

Općenita forma Lefkovitch matrice je:

$$\mathbf{A} = \begin{pmatrix} P_{1,1} + F_{1,1} & F_{1,2} & F_{1,3} & \cdots & F_{1,k} \\ P_{2,1} & P_{2,2} & P_{2,3} & \cdots & P_{2,k} \\ P_{3,1} & P_{3,2} & P_{3,3} & \cdots & P_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{k,1} & P_{k,2} & P_{k,3} & \cdots & P_{k,k} \end{pmatrix}$$

gdje $P_{i,j}$ predstavlja vjerojatnost tranzicije iz stanja j u stanje i, a $F_{i,j}$ predstavlja plodnost stanja j koja doprinosi stanju i.

Ključne razlike u odnosu na Leslie matrice uključuju:

1. **Mogućnost zadržavanja u istom stanju**: dijagonalni elementi $P_{i,i}$ mogu biti pozitivni
2. **Mogućnost kretanja "unazad"**: $P_{i,j}$ može biti pozitivan za $i < j$
3. **Mogućnost "preskakanja" stanja**: $P_{i,j}$ može biti pozitivan za $|i - j| > 1$
4. **Fleksibilniji reproduktivni obrazac**: $F_{i,j}$ može biti pozitivan za bilo koje i, j

Analiza osjetljivosti i elastičnosti

Jedne od najvažnijih primjena matričnih populacijskih modela su analize osjetljivosti i elastičnosti, koje kvantificiraju kako promjene u demografskim parametrima utječu na populacijski rast.

Osjetljivost dominantne vlastite vrijednosti na promjene u elementu matrice $a_{i,j}$ definira se kao:

$$s_{i,j} = \frac{\partial \lambda_1}{\partial a_{i,j}}$$

Koristeći teoriju perturbacije vlastite vrijednosti, osjetljivost se može izračunati kao:

$$s_{i,j} = \frac{v_{1i} w_{1j}}{\langle \mathbf{v}_1, \mathbf{w}_1 \rangle}$$

gdje $\langle \mathbf{v}_1, \mathbf{w}_1 \rangle = \mathbf{v}_1^T \mathbf{w}_1$ je skalarni produkt lijevog i desnog vlastitog vektora.

Elastičnost predstavlja proporcionalnu osjetljivost:

$$e_{i,j} = \frac{a_{i,j}}{\lambda_1} \frac{\partial \lambda_1}{\partial a_{i,j}} = \frac{a_{i,j}}{\lambda_1} s_{i,j}$$

Elastičnosti zadovoljavaju važno svojstvo:

$$\sum_{i,j} e_{i,j} = 1$$

što omogućava izravnu usporedbu relativne važnosti različitih demografskih parametara.

Dekomozicija demografskih doprinosa

Dominantna vlastita vrijednost može se dekomponirati u doprinose različitih demografskih procesa kroz analizu petlje (loop analysis). Za Leslie matricu, osnovne demografske petlje su:

1. **Reproduktivne petlje**: $F_i \cdot \frac{P_1 P_2 \cdots P_{i-1}}{\lambda_1^i}$ i
2. **Petlje preživljavanja**: $\frac{P_1 P_2 \cdots P_{k-1}}{\lambda_1^k}$

Suma svih petlji mora biti jednaka 1:

$$1 = \sum_{i=1}^{k} F_i \cdot \frac{P_1 P_2 \cdots P_{i-1}}{\lambda_1^i} + \frac{P_1 P_2 \cdots P_{k-1}}{\lambda_1^k}$$

Ova dekomozicija, poznata kao Eulerova jednadžba, povezuje sve demografske parametre s asimptotskom stopom rasta i omogućava razumijevanje relativnih doprinosa različitih putova kroz životni ciklus.

Proširenja i primjene

Matrični pristupi mogu se proširiti na različite načine:

1. **Stohastičnost**: Uvođenje vremenski varijabilnih matrica $\mathbf{A}(t)$ za modeliranje čimbenika okoliša
2. **Density-dependence**: Parametri matrice postaju funkcije veličine populacije

3. **Prostorna struktura**: Mega-matrice koje kombiniraju demografske i dispersalne procese

4. **Međuspecijske interakcije**: Povezani sustavi matrica za različite vrste

Matrični modeli našli su široku primjenu u biologiji zaštite prirode, upravljanju prirodnim resursima, ribolovu, šumarstvu i evolucijskoj biologiji. Njihova kombinacija matematičke rigoroznosti i biološke interpretabilnosti čini ih nezamjenjivim alatima za razumijevanje i upravljanje populacijskim procesima.

Dominantna vlastita vrijednost λ_1, kao ključni parametar koji određuje asimptotsku stopu rasta, ostaje centralna veličina u populacijskoj biologiji, omogućavajući preciznu kvantifikaciju demografskog utjecaja različitih čimbenika na dugoročnu vijabilnost populacija.

2.2 Diferencijalne i diferentne jednadžbe

Kontinuirani vs. diskretni vremenski pristup

Matematički opis populacijske dinamike temelji se na dva komplementarna pristupa koji se razlikuju u tretiranju vremenske varijable: kontinuiranim modelima izraženim diferencijalnim jednadžbama i diskretnim modelima izraženim diferentnim jednadžbama.

Diferencijalne jednadžbe za kontinuirane procese

Diferencijalne jednadžbe koriste se kada promatramo procese koji se odvijaju kontinuirano u vremenu. U ovom pristupu, vrijeme t je kontinuirana varijabla koja može poprimiti bilo koju realnu vrijednost iz određenog intervala. Populacijska veličina $N(t)$ također je kontinuirana funkcija vremena.

Osnovna forma diferencijalne jednadžbe za populacijsku dinamiku je:

$$\frac{dN}{dt} = f(N, t)$$

gdje $\frac{dN}{dt}$ predstavlja trenutačnu stopu promjene populacije u vremenu t, a $f(N, t)$ je funkcija koja opisuje kako ta stopa ovisi o trenutačnoj veličini populacije i vremenu.

Ovaj pristup je prikladan za:

- Organizme s preklapajućim generacijama
- Procese koji se odvijaju kontinuirano (npr. rođenja i smrti mogu nastupiti u bilo kojem trenutku)
- Situacije gdje su demografski procesi relativno brzi u odnosu na vremensku skalu promatranja
- Mikroorganizme, biljke, ili populacije s asinkronim reproduktivnim ciklusima

Diferentne jednadžbe za diskretne procese

Diferentne jednadžbe (također nazivane diskretnim jednadžbama ili jednadžbama razlika) koriste se kada promatramo procese koji se odvijaju u diskretnim vremenskim koracima. Vrijeme se tretira kao niz diskretnih točaka $t = 0, 1, 2, 3, ...$ koji obično predstavljaju sezone, godine, ili generacije.

Osnovna forma diferentne jednadžbe za populacijsku dinamiku je:

$$N_{t+1} = g(N_t, t)$$

gdje N_t predstavlja veličinu populacije u vremenskom koraku t, a N_{t+1} je veličina populacije u sljedećem vremenskom koraku.

Ovaj pristup je prikladan za:

- Organizme s diskretnim, sinkroniziranim reproduktivnim sezonama

- Situacije gdje su generacije jasno odvojene
- Godišnje ili sezonske životne cikluse
- Mnoge insekte, godišnje biljke, ili populacije s izraženom sezonalnosti
- Slučajeve gdje su podaci dostupni samo u diskretnim vremenskim intervalima

Matematička veza između pristupa

Povezanost između kontinuiranih i diskretnih modela može se uspostaviti kroz koncepte kao što su:

Za mala vremenska kašnjenja Δt, diferencijalna jednadžba može se aproksimirati diferentnom jednadžbom:

$$\frac{dN}{dt} \approx \frac{N_{t+\Delta t} - N_t}{\Delta t}$$

što daje:

$$N_{t+\Delta t} \approx N_t + \Delta t \cdot f(N_t, t)$$

Obrnuto, diferentna jednadžba može se povezati s diferencijalnom kroz:

$$\ln\left(\frac{N_{t+1}}{N_t}\right) = r \quad \Rightarrow \quad \frac{N_{t+1}}{N_t} = e^r$$

gdje je r kontinuirana stopa rasta povezana s diskretnom stopom rasta $\lambda = \frac{N_{t+1}}{N_t}$ preko:

$$\lambda = e^r \quad \text{ili} \quad r = \ln(\lambda)$$

Matematički opis populacijske dinamike temelji se na ovim komplementarnim pristupima, pri čemu izbor ovisi o biologiji organizma, vremenskim skalama procesa i dostupnosti podataka. Eksponencijalni i logistički modeli rasta predstavljaju temeljna rješenja koja ilustriraju ključne principe populacijske dinamike u oba okvira.

Kontinuirani eksponencijalni rast

Najjednostavniji model populacijske dinamike pretpostavlja da je trenutačna stopa promjene populacije proporcionalna trenutačnoj veličini populacije. Ovaj model, ursprno formuliran od strane Malthusa, može se izraziti diferencijalnom jednadžbom:

$$\frac{dN}{dt} = rN$$

gdje je $N(t)$ veličina populacije u vremenu t, a r je intrinsična stopa rasta populacije (često nazivana Malthusovim parametrom).

Parametar r ima fundamentalnu biološku interpretaciju. On predstavlja razliku između per capita stope rođenja (b) i per capita stope smrti (d):

$$r = b - d$$

Kada je $r > 0$, populacija rasta; kada je $r < 0$, populacija opada; a kada je $r = 0$, populacija je u ravnoteži.

Za rješavanje ove separabilne diferencijalne jednadžbe, možemo odvojiti varijable:

$$\frac{dN}{N} = r\,dt$$

Integriranjem obje strane:

$$\int_{N_0}^{N(t)} \frac{dN'}{N'} = \int_0^t r\,dt'$$

$$\ln N(t) - \ln N_0 = rt$$

$$\ln\left(\frac{N(t)}{N_0}\right) = rt$$

Exponenciranjem dobivamo rješenje:

$$N(t) = N_0 e^{rt}$$

gdje je $N_0 = N(0)$ početna veličina populacije.

Ovo rješenje opisuje eksponencijalni rast ($r > 0$) ili eksponencijalni pad ($r < 0$). Vrijeme udvostručivanja populacije (doubling time) može se izračunati kao:

$$t_d = \frac{\ln 2}{r}$$

dok je poluživot populacije (half-life) u slučaju opadanja:

$$t_{1/2} = \frac{\ln 2}{|r|}$$

Kontinuirani logistički rast

Eksponencijalni model nerealno pretpostavlja neograničene resurse. Verhulstov logistički model modificira eksponencijalni model uključivanjem density-dependent čimbenika kroz uvođenje nosivosti staništa K:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$$

Ovdje izraz u zagradi $\left(1 - \frac{N}{K}\right)$ predstavlja "otpor okoliša" - kada je $N \ll K$, otpor je minimalan i populacija raste gotovo eksponencijalno; kada se N približava K, otpor postaje značajan i usporava rast.

Logistička jednadžba može se prepisati kao:

$$\frac{dN}{dt} = rN - \frac{r}{K}N^2$$

što jasno pokazuje da je ovo nelinearna diferencijalna jednadžba zbog kvadratnog člana.

Za rješavanje logističke jednadžbe, koristimo separaciju varijabli:

$$\frac{dN}{N(1 - \frac{N}{K})} = r\,dt$$

Lijeva strana zahtijeva parcijalnu dekompoziciju:

$$\frac{1}{N(1 - \frac{N}{K})} = \frac{1}{N\left(\frac{K-N}{K}\right)} = \frac{K}{N(K-N)}$$

Koristeći parcijalne frakcije:

$$\frac{K}{N(K-N)} = \frac{A}{N} + \frac{B}{K-N}$$

Rješavanjem za A i B dobivamo $A = 1$ i $B = 1$, pa:

$$\frac{K}{N(K-N)} = \frac{1}{N} + \frac{1}{K-N}$$

Sada možemo integrirati:

$$\int_{N_0}^{N(t)} \left(\frac{1}{N'} + \frac{1}{K - N'} \right) dN' = \int_0^t r \, dt'$$

$$\ln N(t) - \ln N_0 - \ln(K - N(t)) + \ln(K - N_0) = rt$$

$$\ln \left(\frac{N(t)}{N_0} \cdot \frac{K - N_0}{K - N(t)} \right) = rt$$

$$\ln \left(\frac{N(t)(K - N_0)}{N_0(K - N(t))} \right) = rt$$

Exponenciranjem i rješavanjem za $N(t)$:

$$\frac{N(t)(K - N_0)}{N_0(K - N(t))} = e^{rt}$$

$$N(t)(K - N_0) = N_0(K - N(t))e^{rt}$$

$$N(t)(K - N_0) = N_0 K e^{rt} - N_0 N(t) e^{rt}$$

$$N(t)[(K - N_0) + N_0 e^{rt}] = N_0 K e^{rt}$$

$$N(t) = \frac{N_0 K e^{rt}}{(K - N_0) + N_0 e^{rt}}$$

Dijeljenjem brojnika i nazivnika sa $N_0 e^{rt}$:

$$N(t) = \frac{K}{1 + \frac{K - N_0}{N_0} e^{-rt}}$$

što je standardni oblik logističke funkcije.

Analiza logističke krivulje

Logistička krivulja $N(t)$ ima karakteristična svojstva:

1. **Početno ponašanje**: Za mala t, kada je $e^{-rt} \gg 1$:

$$N(t) \approx \frac{K N_0}{K - N_0} e^{rt} = \frac{N_0}{1 - \frac{N_0}{K}} e^{rt}$$

   što se aproximira eksponencijalnim rastom kada je $N_0 \ll K$.

2. **Asimptotsko ponašanje**: Kada $t \to \infty$, $e^{-rt} \to 0$:

$$\lim_{t \to \infty} N(t) = K$$

3. **Infleksijska točka**: Maksimalna stopa rasta postiže se kada je $\frac{d^2N}{dt^2} = 0$.

   Izračunavanjem druge derivacije:

$$\frac{d^2N}{dt^2} = \frac{d}{dt} \left[rN \left(1 - \frac{N}{K} \right) \right] = r \frac{dN}{dt} \left(1 - \frac{2N}{K} \right)$$

Postavivši $\frac{d^2N}{dt^2} = 0$ i koristeći $\frac{dN}{dt} \neq 0$:

$$1 - \frac{2N}{K} = 0 \Rightarrow N = \frac{K}{2}$$

Infleksijska točka nastupa kada populacija dosegne polovicu nosivosti staništa.

4. **Maksimalna stopa rasta**: U infleksijskoj točki:

$$\left.\frac{dN}{dt}\right|_{N=K/2} = r \cdot \frac{K}{2} \cdot \left(1 - \frac{1}{2}\right) = \frac{rK}{4}$$

Diskretni eksponencijalni rast

Za organizme s diskretnim reproduktivnim sezonama, kontinuirani modeli mogu biti neprikladni. Diskretni model eksponencijalnog rasta može se izraziti kao:

$$N_{t+1} = \lambda N_t$$

gdje je λ konačna stopa rasta (finite rate of increase).
Povezanost između kontinuiranog i diskretnog parametra rasta je:

$$\lambda = e^r$$

odnosno:

$$r = \ln \lambda$$

Iterativno rješenje diskretnog modela daje:

$$N_t = \lambda^t N_0$$

što je diskretni ekvivalent kontinuiranog eksponencijalnog rasta.

Diskretni logistički rast

Diskretna verzija logističkog modela može se formulirati različito, ovisno o pretpostavkama o timing efektima density dependence. Najčešći pristupi su:

1. **Ricker model**:
$$N_{t+1} = N_t e^{r(1-\frac{N_t}{K})}$$
Ovaj model pretpostavlja da density dependence djeluje eksponencijalno na stopu rasta.

2. **Beverton-Holt model**:
$$N_{t+1} = \frac{\lambda N_t}{1 + \frac{(\lambda-1)N_t}{K}}$$
gdje je $\lambda > 1$ maksimalna stopa rasta pri malim abundancijama.

Analiza stabilnosti kontinuiranih sustava

Za analizu stabilnosti logističkog modela, linearizirамo oko ravnotežne točke $N^* = K$. Neka je $n(t) = N(t) - K$ mala perturbacija oko ravnoteže:

$$\frac{dn}{dt} = \frac{d(N-K)}{dt} = \frac{dN}{dt} = r(n+K)\left(1 - \frac{n+K}{K}\right)$$

$$= r(n+K)\left(-\frac{n}{K}\right) = -\frac{r}{K}(n+K)n = -\frac{r}{K}n^2 - rn$$

Za male perturbacije, zanemarujemo kvadratni član:

$$\frac{dn}{dt} \approx -rn$$

što ima rješenje $n(t) = n_0 e^{-rt}$. Budući da je $r > 0$, perturbacije eksponencijalno opadaju, što potvrđuje stabilnost ravnotežne točke $N^* = K$.

Analiza stabilnosti diskretnih sustava

Za Ricker model, ravnotežne točke nalazimo rješavanjem:

$$N^* = N^* e^{r(1-\frac{N^*}{K})}$$

Netrivijalna ravnotežna točka je $N^* = K$ (dobiva se iz $r(1 - \frac{N^*}{K}) = 0$).

Za analizu lokalne stabilnosti, lineariziramo oko $N^* = K$:

$$f(N) = Ne^{r(1-\frac{N}{K})}$$

$$f'(N) = e^{r(1-\frac{N}{K})} + N \cdot e^{r(1-\frac{N}{K})} \cdot \left(-\frac{r}{K}\right)$$

$$f'(K) = e^0 + K \cdot e^0 \cdot \left(-\frac{r}{K}\right) = 1 - r$$

Ravnotežna točka je:

- **Stabilna** ako $|f'(K)| < 1$, tj. $|1 - r| < 1$, što daje $0 < r < 2$
- **Nestabilna** ako $|f'(K)| > 1$, tj. $r > 2$ ili $r < 0$

Bifurkacije i kaotično ponašanje

Ricker model pokazuje bogato dinamičko ponašanje ovisno o vrijednosti parametra r:

1. **Za** $0 < r < 2$: Monotonska konvergencija prema K
2. **Za** $r = 2$: Granična stabilnost
3. **Za** $2 < r < 2.526...$: Oscilirajuća konvergencija prema K
4. **Za** $r = 2.526...$: Prvo udvostručenje periode (period-doubling bifurcation)
5. **Za** $2.526... < r < 2.692...$: Ciklus periode 2
6. **Za** $r > 2.692...$: Kaskada udvostručivanja perioda vodeći u kaos

Kaotično ponašanje karakterizira pozitivni Lyapunovov eksponent:

$$\lambda_L = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \ln |f'(N_t)|$$

Za Ricker model:

$$\lambda_L = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \ln |1 - r(1 - \frac{N_t}{K})|$$

Pozitivna vrijednost λ_L indicira kaotično ponašanje s eksponencijalnim razilaženjima bliskih trajektorija.

Beverton-Holt model i kontrastno ponašanje

Za razliku od Ricker modela, Beverton-Holt model ne pokazuje kaotično ponašanje. Ravnotežna točka $N^* = K$ je globalno stabilna za sve $\lambda > 1$.

Linearizacija oko ravnoteže:

$$g(N) = \frac{\lambda N}{1 + \frac{(\lambda-1)N}{K}}$$

$$g'(N) = \frac{\lambda}{1 + \frac{(\lambda-1)N}{K}} - \frac{\lambda N \cdot \frac{\lambda-1}{K}}{(1 + \frac{(\lambda-1)N}{K})^2}$$

$$g'(K) = \frac{\lambda}{1 + (\lambda - 1)} - \frac{\lambda K \cdot \frac{\lambda-1}{K}}{(1 + (\lambda - 1))^2} = \frac{\lambda}{\lambda} - \frac{\lambda(\lambda - 1)}{\lambda^2} = 1 - \frac{\lambda - 1}{\lambda} = \frac{1}{\lambda}$$

Budući da je $\lambda > 1$, imamo $|g'(K)| = \frac{1}{\lambda} < 1$, što garantira lokalnu stabilnost.

Usporedba kontinuiranih i diskretnih modela

Kontinuirani i diskretni pristupi mogu davati kvalitativno različite rezultate:

1. **Kontinuirani logistički model**: Uvijek monotonski konvergira prema nosivosti staništa
2. **Diskretni modeli**: Mogu pokazivati oscilacije, periodičko ponašanje, ili kaos

Ova razlika nastaje zbog različitog tretiranja vremenske skale demografskih procesa. Kontinuirani modeli pretpostavljaju trenutačne demografske odgovore, dok diskretni modeli mogu uhvatiti kašnjenja i prekoračenja (overshoots) karakteristične za organizme s diskretnim generacijama.

Izbor između kontinuiranih i diskretnih modela ovisi o:

- **Biologiji organizma**: Preklapajuće vs. diskretne generacije
- **Vremenskoj skali procesa**: Brzi vs. spori demografski procesi
- **Dostupnosti podataka**: Kontinuirani monitoring vs. godišnji surveyji
- **Ciljevima modeliranja**: Kvalitativno razumijevanje vs. kvantitativno predviđanje

Kombinacija teorijskih uvida iz oba pristupa omogućava dublje razumijevanje kompleksnosti populacijske dinamike i pomoć u donošenju informiranih odluka u ekološkom upravljanju i biologiji zaštite prirode.

2.3 Vjerojatnost, statistika i stohastičnost

2.3.1 Uvod u stohastičnost u ekološkim sustavima

Ekološki sustavi su inherentno nepredvidljivi zbog brojnih čimbenika koji utječu na populacije i zajednice. **Stohastičnost** se odnosi na slučajnu varijabilnost koja se javlja u ekološkim procesima i može se klasificirati u tri glavne kategorije:

1. **Demografska stohastičnost** – slučajne varijacije u rađanju, smrti i reprodukciji individualnih organizama
2. **Okolišna stohastičnost** – slučajne varijacije u uvjetima okoliša
3. **Katastrofična stohastičnost** – rijetki, ali ekstremni događaji

2.3.2 Osnovni koncepti vjerojatnosti

Vjerojatnosne distribucije

Neka je X slučajna varijabla koja opisuje neki ekološki parametar. Funkcija vjerojatnosne gustoće $f(x)$ mora zadovoljiti:

$$\int_{-\infty}^{\infty} f(x)\,dx = 1 \tag{2.1}$$

Za diskretne slučajne varijable, funkcija vjerojatnosne mase $P(X = x_i)$ mora zadovoljiti:

$$\sum_i P(X = x_i) = 1 \tag{2.2}$$

Očekivana vrijednost i varijabilnost

Očekivana vrijednost (matematičko očekivanje):

$$E[X] = \mu = \int_{-\infty}^{\infty} x f(x)\,dx \tag{2.3}$$

Varijanca:

$$\mathrm{Var}(X) = \sigma^2 = E[(X-\mu)^2] = E[X^2] - (E[X])^2 \tag{2.4}$$

Standardna devijacija:

$$\sigma = \sqrt{\mathrm{Var}(X)} \tag{2.5}$$

Koeficijent varijacije

U ekologiji često koristimo **koeficijent varijacije** (CV) za mjerenje relativne varijabilnosti:

$$CV = \frac{\sigma}{\mu} \tag{2.6}$$

2.3.3 Demografska stohastičnost

Binomni model preživljavanja

Pretpostavimo da imamo populaciju od N jedinki, gdje svaka jedinka ima vjerojatnost preživljavanja s. Broj preživjelih jedinki S slijedi binomnu distribuciju:

$$P(S = k) = \binom{N}{k} s^k (1-s)^{N-k} \tag{2.7}$$

Očekivana vrijednost: $E[S] = Ns$
Varijanca: $\mathrm{Var}(S) = Ns(1-s)$

Poissonov model rođenja

Ako je stopa rođenja λ konstantna kroz vrijeme, broj rođenih jedinki u vremenu t slijedi Poissonovu distribuciju:

$$P(X = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \tag{2.8}$$

Očekivana vrijednost: $E[X] = \lambda t$
Varijanca: $\mathrm{Var}(X) = \lambda t$

Utjecaj demografske stohastičnosti na male populacije

Za malu populaciju veličine N, demografska stohastičnost može se aproksimirati normalnom distribucijom:

$$N(t+1) \sim \mathcal{N}(\mu N(t), \sigma^2 N(t)) \tag{2.9}$$

gdje je μ očekivana stopa rasta po jedinki, a σ^2 varijanca demografskih procesa.

Vjerojatnost izumiranja za populaciju s početnom veličinom N_0 i negativnom stopom rasta $r < 0$:

$$P(\text{izumiranje}) = \left(\frac{\sigma^2}{2|r|N_0}\right)^{2|r|/\sigma^2} \tag{2.10}$$

2.3.4 Okolišna stohastičnost

Model s okolišnom stohastičnošću

Stopa rasta populacije varira stohastički ovisno o uvjetima okoliša:

$$\frac{dN}{dt} = r(t)N(t) \tag{2.11}$$

gdje je $r(t)$ stohastički proces. Često pretpostavljamo da je $r(t)$ bijeli šum:

$$r(t) = \mu + \sigma\xi(t) \tag{2.12}$$

gdje je $\xi(t)$ standardni bijeli šum s $E[\xi(t)] = 0$ i $E[\xi(t)\xi(s)] = \delta(t-s)$.

Geometrijski Brownov pokret

Rješenje stohastičke diferencijalne jednadžbe:

$$dN = \mu N\,dt + \sigma N\,dW \tag{2.13}$$

gdje je dW Wienerov proces, daje:

$$N(t) = N_0 \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right] \tag{2.14}$$

Očekivana vrijednost: $E[N(t)] = N_0 e^{\mu t}$
Varijanca: $\text{Var}(N(t)) = N_0^2 e^{2\mu t}(e^{\sigma^2 t} - 1)$

Logistički model s okolišnom stohastičnošću

$$dN = rN\left(1 - \frac{N}{K}\right)dt + \sigma N\,dW \tag{2.15}$$

Stacionarna distribucija (kada postoji) ima oblik:

$$\pi(N) \propto N^{2r/\sigma^2 - 1}\left(1 - \frac{N}{K}\right)^{2r/\sigma^2 - 1} \exp\left(-\frac{2rN}{\sigma^2 K}\right) \tag{2.16}$$

2.3.5 Katastrofična stohastičnost

Poissonov model katastrofa

Katastrofe se javljaju prema Poissonovom procesu s intenzitetom λ. Vjerojatnost da se dogodi k katastrofa u vremenu t:

$$P(N_t = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \tag{2.17}$$

Model s katastrofičnim smanjenjem

Kombinacija eksponencijalnog rasta s povremenim katastrofičnim smanjenjem:

$$N(t^+) = \alpha N(t^-) \tag{2.18}$$

gdje je $\alpha \in (0, 1)$ faktor preživljavanja nakon katastrofe.
Srednja stopa rasta u prisutnosti katastrofa:

$$r_{\text{eff}} = r - \lambda(1 - E[\alpha]) \tag{2.19}$$

gdje je $E[\alpha]$ očekivana vrijednost faktora preživljavanja.

2.3.6 Maksimalna vjerojatnost (Maximum Likelihood)

Likelihood funkcija

Za skup nezavisnih opažanja $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ iz distribucije s parametrima $\boldsymbol{\theta}$, likelihood funkcija je:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i|\boldsymbol{\theta}) \tag{2.20}$$

Log-likelihood

Često je lakše raditi s logaritmom likelihood funkcije:

$$\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(x_i|\boldsymbol{\theta}) \tag{2.21}$$

Maksimalna vjerojatnost procjena (MLE)

Procjenjeni parametri su oni koji maksimiziraju likelihood funkciju:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{2.22}$$

Uvjeti prvog reda:

$$\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_j} = 0, \quad j = 1, 2, \ldots, p \tag{2.23}$$

Primjer: Procjena stope rasta

Za eksponencijalni model populacije $N(t) = N_0 e^{rt}$ s Gaussovim šumom:

$$N_{\text{obs}}(t_i) = N_0 e^{rt_i} + \epsilon_i \tag{2.24}$$

gdje je $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.
Log-likelihood funkcija:

$$\ell(r, N_0, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(N_{\text{obs}}(t_i) - N_0 e^{rt_i})^2 \tag{2.25}$$

Asimptotska svojstva MLE

Za velike uzorke, MLE je:

- **Konzistentan**: $\hat{\boldsymbol{\theta}} \to \boldsymbol{\theta}_0$ kada $n \to \infty$
- **Asimptotski normalan**: $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \to \mathcal{N}(0, \mathbf{I}^{-1})$
- **Asimptotski efikasan**: postiže Cramér-Rao donju granicu

Fisherova informacijska matrica

$$\mathbf{I}_{jk} = -E\left[\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial\theta_j \partial\theta_k}\right] \tag{2.26}$$

Standardne greške procjena: $SE(\hat{\theta}_j) = \sqrt{[\mathbf{I}^{-1}]_{jj}}$

2.3.7 Bayesovska statistika

Bayesov teorem

$$P(\boldsymbol{\theta}|\mathbf{x}) = \frac{P(\mathbf{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{x})} \tag{2.27}$$

gdje je:

- $P(\boldsymbol{\theta}|\mathbf{x})$ – **posteriorni** razočetak
- $P(\mathbf{x}|\boldsymbol{\theta})$ – **likelihood**
- $P(\boldsymbol{\theta})$ – **priorni** razočetak
- $P(\mathbf{x})$ – **marginalna vjerojatnost** (normalizacijska konstanta)

Konjugirani priorovi

Za Poissonovu distribuciju s Gamma priorom:
Prior: $\theta \sim \text{Gamma}(\alpha, \beta)$
Likelihood: $x_i|\theta \sim \text{Poisson}(\theta)$
Posterior: $\theta|\mathbf{x} \sim \text{Gamma}\left(\alpha + \sum x_i, \beta + n\right)$

Kredibilni intervali

95% kredibilni interval za parametar θ:

$$P(\theta_L \leq \theta \leq \theta_U|\mathbf{x}) = 0.95 \tag{2.28}$$

Markov Chain Monte Carlo (MCMC)

Metropolis-Hastings algoritam

1. Započni s početnom vrijednošću $\theta^{(0)}$
2. Za $t = 1, 2, \ldots$:
   - Predloži novu vrijednost: $\theta^* \sim q(\theta^*|\theta^{(t-1)})$
   - Računaj omjer:
   $$\alpha = \min\left(1, \frac{P(\theta^*|\mathbf{x})q(\theta^{(t-1)}|\theta^*)}{P(\theta^{(t-1)}|\mathbf{x})q(\theta^*|\theta^{(t-1)})}\right) \tag{2.29}$$
   - Postavi:
   $$\theta^{(t)} = \begin{cases}\theta^* & \text{s vjerojatnosti } \alpha \\ \theta^{(t-1)} & \text{inače}\end{cases} \tag{2.30}$$

Gibbs sampling Za multidimenzionalne probleme, uzorkuj svaku komponentu uvjetno na ostale:

$$\theta_j^{(t)} \sim P(\theta_j|\theta_1^{(t)}, \ldots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \ldots, \theta_p^{(t-1)}, \mathbf{x}) \tag{2.31}$$

2.3.8 Primjeri u ekološkom modeliranju

Procjena parametara logističkog modela

Za logistički model:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) \tag{2.32}$$

s opažanjima $N_1, N_2, \ldots, N_n$ u vremenima $t_1, t_2, \ldots, t_n$.

Likelihood za kontinuirani model s Gaussovim šumom:

$$L(r, K, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(N_i - N(t_i; r, K))^2}{2\sigma^2}\right) \tag{2.33}$$

Analiza preživljavanja u ekologiji

Za Kaplan-Meier procjenitelj funkcije preživljavanja:

$$\hat{S}(t) = \prod_{t_i \leq t}\left(1 - \frac{d_i}{n_i}\right) \tag{2.34}$$

gdje je d_i broj smrti u vremenu t_i, a n_i broj jedinki pod rizikom.

Bayesova analiza metapopulacije

Za Levinsov metapopulacijski model:

$$\frac{dp}{dt} = cp(1-p) - ep \tag{2.35}$$

s priorom na parametrima:

- $c \sim \text{Gamma}(\alpha_c, \beta_c)$ (stopa kolonizacije)
- $e \sim \text{Gamma}(\alpha_e, \beta_e)$ (stopa izumiranja)

Posteriorni razočetak se računa numerički pomoću MCMC metoda.

2.3.9 Model selekcija i usporedba

Akaike Information Criterion (AIC)

$$AIC = 2k - 2\ell(\hat{\boldsymbol{\theta}}) \tag{2.36}$$

gdje je k broj parametara, a $\ell(\hat{\boldsymbol{\theta}})$ maksimalna log-likelihood vrijednost.

Bayesian Information Criterion (BIC)

$$BIC = k\log(n) - 2\ell(\hat{\boldsymbol{\theta}}) \tag{2.37}$$

Bayes faktori

Za usporedbu modela M_1 i M_2:

$$BF_{12} = \frac{P(\mathbf{x}|M_1)}{P(\mathbf{x}|M_2)} \tag{2.38}$$

Interpretacija:

- $BF_{12} > 10$: snažna podrška za M_1
- $BF_{12} > 100$: odlučujuća podrška za M_1

2.3.10 Propagacija nesigurnosti

Monte Carlo simulacija

Za model $y = f(\mathbf{x})$ gdje je $\mathbf{x}$ vektor parametara s poznatim distribucijama:

1. Generiraj N uzoraka $\mathbf{x}^{(i)}$ iz zajedničke distribucije
2. Računaj $y^{(i)} = f(\mathbf{x}^{(i)})$ za $i = 1, \ldots, N$
3. Analiziraj empirijsku distribuciju $\{y^{(i)}\}$

Delta metoda

Za funkciju $g(\boldsymbol{\theta})$ i MLE $\hat{\boldsymbol{\theta}}$:

$$\sqrt{n}(g(\hat{\boldsymbol{\theta}}) - g(\boldsymbol{\theta}_0)) \to \mathcal{N}(0, \nabla g^T \mathbf{I}^{-1} \nabla g) \tag{2.39}$$

gdje je ∇g gradijent funkcije g.

Bootstrapping

1. Uzorkuj s vraćanjem iz originalnih podataka
2. Računaj statistiku od interesa za svaki bootstrap uzorak
3. Konstruiraj empirijsku distribuciju bootstrap statistika

Bias-corrected and accelerated (BCa) bootstrap interval:

$$\left[\hat{F}^{-1}(\alpha_1), \hat{F}^{-1}(\alpha_2)\right] \tag{2.40}$$

gdje su α_1 i α_2 korigirane vjerojatnosti.

2.3.11 Ključne točke za praktičnu primjenu

1. Uvijek testiraj pretpostavke o distribucijama
2. Koristi dijagnostičke grafove za provjeru modela
3. Kvantificiraj i prenesi nesigurnost u konačne rezultate
4. Kombiniraj različite pristupe za robusne zaključke
5. Dokumentiraj sve pretpostavke i ograničenja analíze

2.4 Računalni alati

2.4.1 Uvod u računalne alate za ekološko modeliranje

Moderne metode ekološkog modeliranja uvelike ovise o računalnim alatima koji omogućavaju implementaciju složenih matematičkih modela, analizu velikih skupova podataka i vizualizaciju rezultata. Četiri glavna alata koji dominiraju u ekološkom modeliranju su R, Python, MATLAB i NetLogo. Svaki od njih ima specifične prednosti ovisno o vrsti modeliranja i analitičkim potrebama.

2.4.2 R programski jezik i okruženje

Uvod u R

R je *open-source* programski jezik i okruženje posebno dizajnirano za statističku analizu i grafiku. Razvila ga je R Core Team, a temelji se na S programskom jeziku. R je postao dominantan alat u ekološkoj statistici zbog svoje fleksibilnosti, opsežne biblioteke paketa i snažne zajednice korisnika.

Prednosti R-a za ekološko modeliranje

- **Specijalizirani paketi**: Preko 18,000 paketa na CRAN repozitoriju
- **Statistička snaga**: Ugrađene funkcije za naprednu statistiku
- **Grafike**: Iznimno fleksibilna grafička mogućnosti
- **Reproducibilnost**: R Markdown i integracija s Git
- **Zajednica**: Aktivna zajednica ekologa i statističara

Ključni paketi za ekološko modeliranje

Osnovni statistički paketi

- `stats`: Osnovne statistike i linearna regresija
- `nlme`: Nelinearna mješovita modeli
- `lme4`: Linearna mješoviti modeli s random efektima
- `mgcv`: Generalizirani aditivni modeli (GAM)
- `MASS`: Moderna primjenjena statistika

Populacijska dinamika i demografija

- `popbio`: Analiza populacijskih matrica
- `demogR`: Demografska analiza
- `Rcompadre`: Baza demografskih podataka
- `lefko3`: Analize životnog ciklusa
- `IPMpack`: Integral Projection Models

Prostorno ekološko modeliranje

- `dismo`: Modeliranje distribucije vrsta
- `raster`: Manipulacija raster podataka
- `sp`: Prostorni podaci
- `sf`: Simple Features za prostorne podatke
- `rgdal`: Geospatial Data Abstraction Library
- `biomod2`: Ansambl modeliranje bioraznolikosti

Ekološke mreže i zajednice

- `vegan`: Analiza ekoloških zajednica
- `igraph`: Analiza mreža
- `bipartite`: Bipartitne ekološke mreže
- `foodweb`: Hranidbene mreže
- `NetIndices`: Mrežni indeksi

Bayesovska analiza

- `MCMCglmm`: MCMC za generalizirane linearne modele
- `rstanarm`: Bayesova regresija putem Stan-a
- `brms`: Bayesova regresija s Stan backend
- `R2jags`: Interface za JAGS
- `nimble`: MCMC algoritmi

Primjer koda: Leslie matrica u R-u

Listing 2.1: Implementacija Leslie matrice

```
# Definiranje Leslie matrice
# F - fertility rates, S - survival rates
create_leslie <- function(F, S) {
       n <- length(F)
       L <- matrix(0, nrow = n, ncol = n)

       # Prvi red: fertility rates
       L[1, ] <- F

       # Subdiagonala: survival rates
```

```
        for(i in 2:n) {
                L[i, i-1] <- S[i-1]
        }

        return(L)
}

# Parametri za primjer
fertility <- c(0, 0.3, 1.2, 1.5, 0.8)
survival <- c(0.6, 0.7, 0.8, 0.3)

# Kreiranje matrice
L <- create_leslie(fertility, survival)

# čPoetna populacija
n0 <- c(100, 80, 60, 40, 20)

# Simulacija kroz vrijeme
simulate_population <- function(L, n0, t_max = 50) {
        n_age <- length(n0)
        N <- matrix(NA, nrow = n_age, ncol = t_max + 1)
        N[, 1] <- n0

        for(t in 1:t_max) {
                N[, t + 1] <- L %*% N[, t]
        }

        return(N)
}

# Pokretanje simulacije
results <- simulate_population(L, n0, 50)

# čRaunanje ukupne populacije
total_pop <- colSums(results)

# Analiza asimptotskog šponaanja
eigen_analysis <- eigen(L)
lambda <- Re(eigen_analysis$values[1])
stable_age_dist <- Re(eigen_analysis$vectors[, 1])
stable_age_dist <- stable_age_dist / sum(stable_age_dist)

cat("Asimptotska stopa rasta (lambda):", lambda, "\n")
cat("Stabilna dobna distribucija:", stable_age_dist, "\n")
```

Upravljanje paketima u R-u

Listing 2.2: Instalacija paketa iz CRAN-a

CRAN repozitorij

```
# Instalacija čpojedinanog paketa
install.packages("dismo")

# Instalacija švie paketa odjednom
packages <- c("vegan", "mgcv", "lme4", "ggplot2")
install.packages(packages)
```

```
# Provjera dostupnih verzija
available.packages()[c("dismo", "vegan"), ]
```

Listing 2.3: Instalacija iz GitHub-a

GitHub repozitoriji

```
# Instalacija devtools
install.packages("devtools")

# Instalacija paketa iz GitHub-a
devtools::install_github("ropensci/spocc")
devtools::install_github("user/package", ref = "branch_name")
```

Listing 2.4: Reproducibilno okruženje s renv

Upravljanje verzijama s renv

```
# Inicijalizacija projekta
renv::init()

# Snapshot trenutnog stanja
renv::snapshot()

# ćVraanje na prethodno stanje
renv::restore()

# žAuriranje paketa
renv::update()
```

2.4.3 Python programski jezik

Uvod u Python za ekologiju

Python je *general-purpose* programski jezik koji je postao iznimno popularan u znanstvenoj zajednici zbog svoje čitljivosti, fleksibilnosti i opsežnog ekosustava znanstvenih biblioteka. Za ekološko modeliranje, Python nudi prednosti u integraciji s drugim sustavima, obradi velikih podataka i strojnom učenju.

Ključne biblioteke za ekološko modeliranje

Znanstvene računalne biblioteke

- `numpy`: Numeričko računanje i rad s nizovima
- `scipy`: Znanstveni algoritmi i optimizacija
- `pandas`: Manipulacija i analiza podataka
- `matplotlib`: Osnovna vizualizacija
- `seaborn`: Statistička vizualizacija
- `plotly`: Interaktivna vizualizacija

Strojno učenje i statistika

- `scikit-learn`: Opće potrebe strojnog učenja
- `statsmodels`: Statistički modeli
- `pymc`: Bayesovska statistika
- `tensorflow/pytorch`: Duboko učenje
- `xgboost`: Gradient boosting

Prostorni podaci i GIS

- `geopandas`: Prostorni podaci
- `rasterio`: Raster podaci
- `shapely`: Geometrijske operacije
- `fiona`: Čitanje/pisanje prostornih podataka
- `folium`: Interaktivne karte

Specijalizirane ekološke biblioteke

- `ecohydro`: Ekohidrološki modeli
- `scikit-bio`: Bioinformatika
- `biopython`: Molekularna biologija
- `ete3`: Filogenetska analiza

Primjer koda: Lotka-Volterra model u Pythonu

Listing 2.5: Lotka-Volterra predator-prey model (dio 1)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

class LotkaVolterra:
"""
Klasa za Lotka-Volterra predator-prey model
"""

def __init__(self, r, a, e, m):
"""
Parametri:
r - stopa rasta plijena
a - stopa napada
e - efikasnost konverzije
m - stopa smrti predatora
"""
self.r = r # intrinsic growth rate of prey
self.a = a # attack rate
self.e = e # conversion efficiency
self.m = m # predator death rate

def equations(self, t, z):
"""
```

```
        Lotka-Volterra diferencijalne žjednadbe
        """
        N, P = z # prey, predator

        dN_dt = self.r * N - self.a * N * P
        dP_dt = self.e * self.a * N * P - self.m * P

        return [dN_dt, dP_dt]

        def simulate(self, initial_conditions, t_span, t_eval=None):
        """
        Simulacija modela
        """
        if t_eval is None:
        t_eval = np.linspace(t_span[0], t_span[1], 1000)

        solution = solve_ivp(
        self.equations,
        t_span,
        initial_conditions,
        t_eval=t_eval,
        dense_output=True
        )

        return solution
```

Upravljanje okruženjima u Pythonu

Listing 2.6: Conda environment management

Conda upravljanje paketima

```
        # Kreiranje novog žokruenja
        conda create -n ecology python=3.9

        # Aktivacija žokruenja
        conda activate ecology

        # Instalacija paketa
        conda install numpy scipy pandas matplotlib
        conda install -c conda-forge geopandas

        # Export žokruenja
        conda env export > environment.yml

        # Kreiranje žokruenja iz filea
        conda env create -f environment.yml

        # Lista svih žokruenja
        conda env list
```

Listing 2.7: pip package management

pip i virtualenv

```
        # Kreiranje virtualnog žokruenja
        python -m venv ecology_env
```

```
# Aktivacija (Linux/Mac)
source ecology_env/bin/activate

# Aktivacija (Windows)
ecology_env\Scripts\activate

# Instalacija paketa
pip install numpy scipy pandas matplotlib

# Generiranje requirements.txt
pip freeze > requirements.txt

# Instalacija iz requirements.txt
pip install -r requirements.txt
```

2.4.4 MATLAB

Uvod u MATLAB za ekološko modeliranje

MATLAB je komercijalni programski jezik i okruženje optimizirano za numeričko računanje i matrične operacije. Iako je komercijalan, MATLAB nudi iznimno snažne alate za simulaciju dinamičkih sustava, numeričku optimizaciju i vizualizaciju, što ga čini popularnim u inženjerskim pristupima ekološkom modeliranju.

Prednosti MATLAB-a

- **Numerička stabilnost**: Visoko optimizirani algoritmi
- **Simulink**: Grafičko modeliranje dinamičkih sustava
- **Toolboxovi**: Specijalizirani alati za različite domene
- **Paralelno računanje**: Ugrađena podrška za GPU i klaster računanje
- **Integracija**: Povezivanje s drugim sustavima i hardware

Relevantni toolboxovi za ekologiju

- **Statistics and Machine Learning Toolbox**: Statistička analiza
- **Optimization Toolbox**: Numerička optimizacija
- **Parallel Computing Toolbox**: Paralelno izvršavanje
- **Mapping Toolbox**: GIS i kartografija
- **Image Processing Toolbox**: Analiza slika (daljinska istraživanja)
- **Global Optimization Toolbox**: Globalna optimizacija

2.4.5 GNU Octave

Uvod u GNU Octave

GNU Octave je besplatna, open-source alternativa MATLAB-u s visokom sintaksnom kompatibilnošću. Razvijan od 1988. godine, Octave omogućava izvršavanje većine MATLAB koda bez promjena, što ga čini idealnim za prelazak s komercijalnih na open-source alate ili za institucije s ograničenim budgetom.

Prednosti Octave-a za ekološko modeliranje

- **MATLAB kompatibilnost**: 95%+ kompatibilnost sintakse
- **Besplatnost**: Potpuno besplatan za akademsku i komercijalnu upotrebu
- **Cross-platform**: Dostupan za Windows, Mac, Linux
- **Aktivna zajednica**: Redovita ažuriranja i podrška
- **Proširivost**: Mogućnost dodavanja C++ funkcija

Ključni paketi za ekološko modeliranje

Osnovni paketi

- `statistics`: Statistička analiza i distribucije
- `optim`: Optimizacija i fitting algoritmi
- `signal`: Obrada signala i filtriranje
- `image`: Osnovna obrada slika
- `io`: Input/output operacije za različite formate

Numeričko rješavanje

- `ode`: Dodatni rješavači diferencijalnih jednadžbi
- `control`: Teorija upravljanja i analiza sustava
- `symbolic`: Simboličko računanje
- `parallel`: Paralelno izvršavanje

Primjer koda: Populacijski model u Octave

Listing 2.8: Stohastički populacijski model u Octave

```
function stochastic_population_model()
% čStohastiki populacijski model s šokolinim varijacijama

% Parametri
params = struct();
params.r_mean = 0.1; % srednja stopa rasta
params.r_std = 0.05; % varijabilnost stope rasta
params.K = 1000; % nosivost šstanita
params.sigma_demographic = 0.1; % demografska čstohastinost
params.dt = 0.1; % vremenski korak
params.t_max = 50; % maksimalno vrijeme
params.n_simulations = 100; % broj simulacija

% Vremenska žmrea
t = 0:params.dt:params.t_max;
n_steps = length(t);

% Matrica za pohranu rezultata
N = zeros(params.n_simulations, n_steps);

% čPoetna populacija
N0 = 50;
```

```
% Monte Carlo simulacije
for sim = 1:params.n_simulations
N(sim, 1) = N0;

for i = 2:n_steps
% Trenutna populacija
N_current = N(sim, i-1);

% šOkolina čstohastinost (varijacija u r)
r_current = params.r_mean + params.r_std * randn();

% čDeterministiki dio
dN_det = r_current * N_current * (1 - N_current / params.K);

% Demografska čstohastinost
if N_current > 0
dN_stoch = params.sigma_demographic * sqrt(N_current) * randn();
else
dN_stoch = 0;
end

% žAuriranje populacije
N(sim, i) = max(0, N_current + (dN_det + dN_stoch) * params.dt);

% Provjera izumiranja
if N(sim, i) < 1
N(sim, i:end) = 0;
break;
end
end
end

% Analiza rezultata
analyze_results(t, N, params);

% Vizualizacija
visualize_results(t, N, params);
end

function analyze_results(t, N, params)
% Osnovne statistike
final_populations = N(:, end);
extinctions = sum(final_populations == 0);
extinction_probability = extinctions / params.n_simulations;

fprintf('Analiza čstohastikog populacijskog modela:\n');
fprintf('Broj simulacija: %d\n', params.n_simulations);
fprintf('Vjerojatnost izumiranja: %.3f\n', extinction_probability);

% Statistike za žpreivjele populacije
survivors = final_populations(final_populations > 0);
if ~isempty(survivors)
fprintf('Srednja čkonana populacija (žpreivjeli): %.2f ± %.2f\n', ...
mean(survivors), std(survivors));
fprintf('Raspon čkonanih populacija: %.1f - %.1f\n', ...
min(survivors), max(survivors));
```

```
end

% Vrijeme do izumiranja
extinction_times = zeros(extinctions, 1);
extinct_count = 0;

for sim = 1:params.n_simulations
extinct_indices = find(N(sim, :) == 0, 1);
if ~isempty(extinct_indices)
extinct_count = extinct_count + 1;
extinction_times(extinct_count) = t(extinct_indices);
end
end

if extinct_count > 0
fprintf('Srednje vrijeme do izumiranja: %.2f ± %.2f\n', ...
mean(extinction_times), std(extinction_times));
end
end

function visualize_results(t, N, params)
% Kreiranje slika
figure('Position', [100, 100, 1200, 600]);

% Subplot 1: Vremenske serije
subplot(2, 3, 1);
plot(t, N(1:min(20, size(N,1)), :)', 'Color', [0.7, 0.7, 0.7]);
hold on;
plot(t, mean(N, 1), 'r-', 'LineWidth', 2);
plot(t, quantile(N, 0.05, 1), 'b--', 'LineWidth', 1);
plot(t, quantile(N, 0.95, 1), 'b--', 'LineWidth', 1);
xlabel('Vrijeme');
ylabel('čVeliina populacije');
title('čStohastike trajektorije');
legend('Simulacije', 'Prosjek', '5-95% kvantili', 'Location', 'best');
grid on;

% Subplot 2: Distribucija čkonanih populacija
subplot(2, 3, 2);
final_pops = N(:, end);
final_pops_nonzero = final_pops(final_pops > 0);

if ~isempty(final_pops_nonzero)
histogram(final_pops_nonzero, 20);
xlabel('čKonana čveliina populacije');
ylabel('Frekvencija');
title('Distribucija čkonanih populacija');
grid on;
end

% Subplot 3: Vjerojatnost žpreivljavanja kroz vrijeme
subplot(2, 3, 3);
survival_prob = zeros(size(t));
for i = 1:length(t)
survival_prob(i) = sum(N(:, i) > 0) / params.n_simulations;
end
```

```
plot(t, survival_prob, 'g-', 'LineWidth', 2);
xlabel('Vrijeme');
ylabel('Vjerojatnost žpreivljavanja');
title('Krivulja žpreivljavanja');
grid on;
ylim([0, 1]);

% Subplot 4: Varijabilnost kroz vrijeme
subplot(2, 3, 4);
cv_through_time = std(N, 1) ./ mean(N, 1);
cv_through_time(isnan(cv_through_time)) = 0;

plot(t, cv_through_time, 'm-', 'LineWidth', 2);
xlabel('Vrijeme');
ylabel('Koeficijent varijacije');
title('Varijabilnost populacije');
grid on;

% Subplot 5: Fazni dijagram (N vs dN/dt)
subplot(2, 3, 5);
% čRaunanje aproksimativnog dN/dt
N_sample = N(1, :); % Uzmi prvu simulaciju kao primjer
dN_dt = diff(N_sample) / params.dt;
N_mid = N_sample(1:end-1);

plot(N_mid, dN_dt, 'ko', 'MarkerSize', 3);
xlabel('N');
ylabel('dN/dt');
title('Fazni dijagram');
grid on;

% Dodaj teorijsku krivulju
N_theory = 0:params.K/100:params.K;
dN_theory = params.r_mean * N_theory .* (1 - N_theory / params.K);
hold on;
plot(N_theory, dN_theory, 'r-', 'LineWidth', 2);
legend('Simulacija', 'Teorijski', 'Location', 'best');

% Subplot 6: Autocorrelation analiza
subplot(2, 3, 6);
% Autocorrelation srednje populacije
mean_pop = mean(N, 1);
mean_pop_detrended = detrend(mean_pop);

% čIzraunaj autokorelaciju
max_lag = min(50, floor(length(mean_pop_detrended)/4));
autocorr_values = zeros(max_lag+1, 1);

for lag = 0:max_lag
if lag == 0
autocorr_values(lag+1) = 1;
else
x1 = mean_pop_detrended(1:end-lag);
x2 = mean_pop_detrended(lag+1:end);
autocorr_values(lag+1) = corr(x1', x2');
end
end
```

```
lags = 0:max_lag;
plot(lags * params.dt, autocorr_values, 'b-o', 'LineWidth', 1.5);
xlabel('Lag (vrijeme)');
ylabel('Autokorelacija');
title('Vremenska autokorelacija');
grid on;

% Dodaj liniju na y=0
hold on;
plot([0, max(lags) * params.dt], [0, 0], 'k--');
end

% Pokretanje simulacije
stochastic_population_model();
```

Instalacija i upravljanje paketima

Listing 2.9: Instalacija GNU Octave

Instalacija Octave-a

```
# Ubuntu/Debian
sudo apt-get install octave octave-doc

# MacOS (Homebrew)
brew install octave

# Windows - download from https://www.gnu.org/software/octave/

# Fedora/CentOS
sudo dnf install octave
```

Listing 2.10: Octave package management

Upravljanje paketima

```
% Lista dostupnih paketa
pkg list

% Instalacija paketa
pkg install -forge statistics
pkg install -forge optim
pkg install -forge signal

% čUitavanje paketa
pkg load statistics

% žAuriranje svih paketa
pkg update

% Uklanjanje paketa
pkg uninstall statistics

% Provjera statusa paketa
pkg describe statistics
```

2.4.6 Python Scientific Computing Stack

SciPy ekosustav za ekološko modeliranje

SciPy ekosustav predstavlja kolekciju Python biblioteka optimiziranih za znanstveno računanje. Ovaj stack čini temelj za većinu znanstvenih Python aplikacija i nudi snažnu alternativu komercijskim sustavima poput MATLAB-a.

Ključne komponente SciPy stack-a

NumPy - Numerička osnova

- **N-dimenzijski nizovi**: Efikasne matrične operacije
- **Broadcasting**: Operacije na nizovima različitih dimenzija
- **Linear algebra**: BLAS/LAPACK optimizacija
- **Random sampling**: Pseudoslučajni broj generatori
- **Fourier transforms**: FFT implementacije

SciPy - Znanstveni algoritmi

- **scipy.integrate**: ODE/PDE rješavači
- **scipy.optimize**: Nelinearna optimizacija
- **scipy.stats**: Statistička distribucije i testovi
- **scipy.interpolate**: Interpolacija i aproksimacija
- **scipy.signal**: Obrada signala

Matplotlib - Vizualizacija

- **2D plotting**: Line plots, scatter plots, heatmaps
- **3D visualization**: Surface plots, volumetric rendering
- **Animation**: Dinamičke vizualizacije
- **Publication quality**: LaTeX podrška, fine-tuning

Napredni primjer: Prostorno-eksplicitni epidemiološki model

Listing 2.11: SIR model s prostornom difuzijom

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from scipy.ndimage import gaussian_filter
import matplotlib.animation as animation
from mpl_toolkits.mplot3d import Axes3D

class SpatialSIRModel:
"""
Prostorno-eksplicitni SIR šepidemioloki model
s difuzijom i heterogenim kontaktnim uzorcima
"""

def __init__(self, grid_size, dx, beta, gamma, D_S, D_I, D_R):
```

```
        """
        Parametri:
        grid_size: tuple (nx, ny) - dimenzije prostorne žmree
        dx: float - prostorna rezolucija
        beta: float ili array - stopa prijenosa
        gamma: float - stopa oporavka
        D_S, D_I, D_R: float - difuzijski koeficijenti
        """
        self.nx, self.ny = grid_size
        self.dx = dx
        self.beta = beta
        self.gamma = gamma
        self.D_S = D_S
        self.D_I = D_I
        self.D_R = D_R

        # Prostorna žmrea
        self.x = np.linspace(0, (self.nx-1)*dx, self.nx)
        self.y = np.linspace(0, (self.ny-1)*dx, self.ny)
        self.X, self.Y = np.meshgrid(self.x, self.y)

        # Laplacian operator za difuziju
        self.laplacian_kernel = np.array([[0, 1, 0],
        [1, -4, 1],
        [0, 1, 0]]) / (dx**2)

        def laplacian_2d(self, field):
        """Compute 2D Laplacian using convolution"""
        from scipy.ndimage import convolve
        return convolve(field, self.laplacian_kernel, mode='constant', cval=0)

        def spatial_sir_system(self, t, state_vector):
        """
        Prostorno-eksplicitni SIR sustav žjednadbi
        """
        # Reshape 1D state vector to 3D array (3 compartments x nx x ny)
        state = state_vector.reshape(3, self.nx, self.ny)
        S, I, R = state[0], state[1], state[2]

        # Ukupna populacija u svakoj ćeliji
        N = S + I + R

        # čSprijei dijeljenje s nulom
        N = np.where(N > 0, N, 1)

        # SIR dynamics
        if isinstance(self.beta, np.ndarray):
        # Heterogeni kontaktni uzorci
        infection_rate = self.beta * S * I / N
        else:
        # Homogeni kontaktni uzorci
        infection_rate = self.beta * S * I / N

        recovery_rate = self.gamma * I

        # Difuzijski operatori
        dS_diffusion = self.D_S * self.laplacian_2d(S)
```

```
        dI_diffusion = self.D_I * self.laplacian_2d(I)
        dR_diffusion = self.D_R * self.laplacian_2d(R)

        # Sustav žjednadbi
        dS_dt = -infection_rate + dS_diffusion
        dI_dt = infection_rate - recovery_rate + dI_diffusion
        dR_dt = recovery_rate + dR_diffusion

        # Return flattened derivative
        return np.array([dS_dt, dI_dt, dR_dt]).flatten()

        def create_initial_conditions(self, total_pop, initial_infected_centers):
        """
        Kreiranje čpoetnih uvjeta s lokaliziranim žšaritima infekcije
        """
        S0 = np.ones((self.nx, self.ny)) * total_pop
        I0 = np.zeros((self.nx, self.ny))
        R0 = np.zeros((self.nx, self.ny))

        # Dodaj žšarita infekcije
        for center_x, center_y, radius, intensity in initial_infected_centers:
        i_center = int(center_x / self.dx)
        j_center = int(center_y / self.dx)

        for i in range(max(0, i_center - radius),
        min(self.nx, i_center + radius + 1)):
        for j in range(max(0, j_center - radius),
        min(self.ny, j_center + radius + 1)):
        distance = np.sqrt((i - i_center)**2 + (j - j_center)**2)
        if distance <= radius:
        infection_fraction = intensity * np.exp(-distance**2 / (radius**2))
        infected_count = min(infection_fraction * S0[i, j], S0[i, j])
        I0[i, j] += infected_count
        S0[i, j] -= infected_count

        return np.array([S0, I0, R0])

        def create_heterogeneous_beta(self, urban_centers, urban_beta, rural_beta):
        """
        Kreiranje heterogene matrice kontaktnih stopa
        """
        beta_map = np.ones((self.nx, self.ny)) * rural_beta

        for center_x, center_y, radius in urban_centers:
        i_center = int(center_x / self.dx)
        j_center = int(center_y / self.dx)

        for i in range(self.nx):
        for j in range(self.ny):
        distance = np.sqrt((i - i_center)**2 + (j - j_center)**2)
        if distance <= radius:
        # Gradijent od urbane do ruralne stope
        urban_fraction = np.exp(-distance**2 / (radius**2))
        beta_map[i, j] = (urban_fraction * urban_beta +
        (1 - urban_fraction) * rural_beta)

        return beta_map
```

```
def simulate(self, initial_state, t_span, t_eval=None):
"""
Pokretanje simulacije
"""
if t_eval is None:
t_eval = np.linspace(t_span[0], t_span[1], 100)

# Flatten initial state
y0 = initial_state.flatten()

# Solve ODE system
solution = solve_ivp(
self.spatial_sir_system,
t_span,
y0,
t_eval=t_eval,
method='RK45',
rtol=1e-6,
atol=1e-8
)

# Reshape solution back to 3D
solution_3d = solution.y.reshape(3, self.nx, self.ny, len(t_eval))

return solution.t, solution_3d

def analyze_epidemic_dynamics(self, t, solution):
"""
Analiza dinamike epidemije
"""
S, I, R = solution[0], solution[1], solution[2]

# Ukupne populacije kroz vrijeme
total_S = np.sum(S, axis=(0, 1))
total_I = np.sum(I, axis=(0, 1))
total_R = np.sum(R, axis=(0, 1))
total_N = total_S + total_I + total_R

# čKljune metrike
peak_infected = np.max(total_I)
peak_time_idx = np.argmax(total_I)
peak_time = t[peak_time_idx]

final_attack_rate = total_R[-1] / total_N[0]

# Prostorna analiza
max_local_infected = np.max(I, axis=2)
spatial_peak_locations = []

for time_idx in range(len(t)):
max_val = np.max(max_local_infected[:, :, time_idx])
if max_val > 0:
max_locations = np.where(max_local_infected[:, :, time_idx] == max_val)
if len(max_locations[0]) > 0:
spatial_peak_locations.append((
time_idx,
```

```
        max_locations[0][0] * self.dx,
        max_locations[1][0] * self.dx,
        max_val
        ))

        return {
               'total_timeseries': (total_S, total_I, total_R),
               'peak_infected': peak_infected,
               'peak_time': peak_time,
               'final_attack_rate': final_attack_rate,
               'spatial_peaks': spatial_peak_locations
        }

        def run_spatial_sir_example():
        """
        Pokretanje primjera prostorno-eksplicitnog SIR modela
        """
        # Parametri modela
        grid_size = (50, 50)
        dx = 1.0 # km
        gamma = 1/14 # oporavak za 14 dana
        D_S = 0.1 # difuzija susceptible
        D_I = 0.05 # difuzija infected (manji zbog karantene)
        D_R = 0.1 # difuzija recovered

        # Kreiranje modela
        model = SpatialSIRModel(grid_size, dx, None, gamma, D_S, D_I, D_R)

        # Heterogena struktura kontakata
        urban_centers = [(15, 15, 8), (35, 35, 6)] # (x, y, radius)
        beta_map = model.create_heterogeneous_beta(
        urban_centers,
        urban_beta=0.5, # visoka stopa u gradovima
        rural_beta=0.1 # niska stopa u ruralnim čpodrujima
        )
        model.beta = beta_map

        # čPoetni uvjeti
        total_population = 1000
        infection_centers = [(15, 15, 3, 0.05), (35, 35, 2, 0.02)] # (x, y, radius,
            intensity)

        initial_state = model.create_initial_conditions(
        total_population,
        infection_centers
        )

        # Simulacija
        t_span = (0, 365) # godina dana
        t_eval = np.linspace(0, 365, 200)

        print("Pokretanje prostorno-eksplicitne SIR simulacije...")
        time, solution = model.simulate(initial_state, t_span, t_eval)

        # Analiza
        results = model.analyze_epidemic_dynamics(time, solution)
```

```
print(f"Vrhunac infekcije: {results['peak_infected']:.0f} na dan {results['
    peak_time']:.1f}")
print(f"čKonana stopa napada: {results['final_attack_rate']:.3f}")

# Vizualizacija
visualize_spatial_sir_results(model, time, solution, results)

def visualize_spatial_sir_results(model, time, solution, results):
"""
Vizualizacija rezultata prostorno-eksplicitnog SIR modela
"""
S, I, R = solution[0], solution[1], solution[2]

# Kreiranje figure
fig = plt.figure(figsize=(15, 10))

# Subplot 1: Vremenske serije ukupnih populacija
ax1 = plt.subplot(2, 3, 1)
total_S, total_I, total_R = results['total_timeseries']

plt.plot(time, total_S, 'b-', label='Susceptible', linewidth=2)
plt.plot(time, total_I, 'r-', label='Infected', linewidth=2)
plt.plot(time, total_R, 'g-', label='Recovered', linewidth=2)
plt.xlabel('Vrijeme (dani)')
plt.ylabel('Broj jedinki')
plt.title('Ukupna dinamika populacije')
plt.legend()
plt.grid(True, alpha=0.3)

# Subplot 2: Prostorna distribucija na vrhuncu
peak_idx = np.argmax(total_I)

ax2 = plt.subplot(2, 3, 2)
im2 = plt.imshow(I[:, :, peak_idx].T, origin='lower',
extent=[0, model.nx*model.dx, 0, model.ny*model.dx],
cmap='Reds', interpolation='bilinear')
plt.colorbar(im2, ax=ax2, label='Infected')
plt.xlabel('X (km)')
plt.ylabel('Y (km)')
plt.title(f'Infected na vrhuncu (dan {time[peak_idx]:.1f})')

# Subplot 3: čKonana prostorna distribucija oporavljenih
ax3 = plt.subplot(2, 3, 3)
im3 = plt.imshow(R[:, :, -1].T, origin='lower',
extent=[0, model.nx*model.dx, 0, model.ny*model.dx],
cmap='Greens', interpolation='bilinear')
plt.colorbar(im3, ax=ax3, label='Recovered')
plt.xlabel('X (km)')
plt.ylabel('Y (km)')
plt.title('čKonana distribucija oporavljenih')

# Subplot 4: Evolucija prostornog centra infekcije
ax4 = plt.subplot(2, 3, 4)

# čRaunanje centra mase infekcije kroz vrijeme
centers_x = []
centers_y = []
```

```
for t_idx in range(len(time)):
I_current = I[:, :, t_idx]
total_infected = np.sum(I_current)

if total_infected > 0:
center_x = np.sum(model.X * I_current) / total_infected
center_y = np.sum(model.Y * I_current) / total_infected
centers_x.append(center_x)
centers_y.append(center_y)
else:
centers_x.append(np.nan)
centers_y.append(np.nan)

# Plot trajectory
valid_indices = ~np.isnan(centers_x)
if np.any(valid_indices):
plt.plot(np.array(centers_x)[valid_indices],
np.array(centers_y)[valid_indices],
'ro-', markersize=3, linewidth=1)
plt.plot(centers_x[0], centers_y[0], 'go', markersize=8, label='čPoetak')
plt.plot(centers_x[peak_idx], centers_y[peak_idx], 'ro', markersize=8, label
    ='Vrhunac')

plt.xlabel('X (km)')
plt.ylabel('Y (km)')
plt.title('Kretanje centra infekcije')
plt.legend()
plt.grid(True, alpha=0.3)

# Subplot 5: Heatmap ukupne incidencije
ax5 = plt.subplot(2, 3, 5)

# Ukupna incidencija = čpoetni S - čkonani S
total_incidence = S[:, :, 0] - S[:, :, -1]

im5 = plt.imshow(total_incidence.T, origin='lower',
extent=[0, model.nx*model.dx, 0, model.ny*model.dx],
cmap='YlOrRd', interpolation='bilinear')
plt.colorbar(im5, ax=ax5, label='Ukupna incidencija')
plt.xlabel('X (km)')
plt.ylabel('Y (km)')
plt.title('Ukupna incidencija')

# Subplot 6: Vremenska evolucija maksimalne lokalne incidencije
ax6 = plt.subplot(2, 3, 6)

max_local_I = np.max(I, axis=(0, 1))
mean_I = np.mean(I, axis=(0, 1))
std_I = np.std(I, axis=(0, 1))

plt.plot(time, max_local_I, 'r-', linewidth=2, label='Maksimalna lokalna')
plt.plot(time, mean_I, 'b-', linewidth=2, label='Srednja')
plt.fill_between(time, mean_I - std_I, mean_I + std_I,
alpha=0.3, color='blue', label='±1 SD')

plt.xlabel('Vrijeme (dani)')
```

```
    plt.ylabel('Broj žzaraenih')
    plt.title('Prostorna varijabilnost infekcije')
    plt.legend()
    plt.grid(True, alpha=0.3)

    plt.tight_layout()
    plt.show()

    # Pokretanje primjera
    if __name__ == "__main__":
    run_spatial_sir_example()
```

2.4.7 Julia programski jezik

Uvod u Julia za ekološko modeliranje

Julia je moderan programski jezik dizajniran za visokoperformantno znanstveno računanje. Kombinira jednostavnost Python-a s brzinom C-a, što ga čini idealnim za računalno zahtjevne ekološke simulacije. Razvijen na MIT-u, Julia posebno je prikladna za numeričke simulacije, optimizaciju i paralelno računanje.

Prednosti Julia za ekologiju

- **Performanse**: Brzina bliska C/Fortran kodu
- **Jednostavnost**: Čitljiva sintaksa slična Python-u i R-u
- **Multiple dispatch**: Fleksibilan sustav funkcija
- **Paralelizam**: Ugrađena podrška za distribuirano računanje
- **Interoperabilnost**: Pozivanje C, Python, R koda
- **Dinamičnost**: REPL okruženje za interaktivno programiranje

Ključni paketi za ekološko modeliranje

Diferencijalne jednadžbe i simulacije

- `DifferentialEquations.jl`: Napredni ODE/SDE/PDE rješavači
- `ModelingToolkit.jl`: Simboličko modeliranje sustava
- `Catalyst.jl`: Reakcijske mreže i biokemijski modeli
- `StochasticDiffEq.jl`: Stohastičke diferencijalne jednadžbe

Optimizacija i fitting

- `Optim.jl`: Optimizacijski algoritmi
- `BlackBoxOptim.jl`: Global optimization
- `DiffEqParamEstim.jl`: Parameter estimation za DE
- `Turing.jl`: Bayesovska analiza

Podatkovni rad i statistika

- `DataFrames.jl`: Manipulacija tabličnih podataka
- `CSV.jl`: Čitanje/pisanje CSV datoteka
- `Statistics.jl`: Osnovne statistike
- `StatsBase.jl`: Napredne statistike
- `Distributions.jl`: Vjerojatnosne distribucije

Vizualizacija

- `Plots.jl`: Opća plotting biblioteka
- `PlotlyJS.jl`: Interaktivne web vizualizacije
- `Makie.jl`: Visokokvalitetna grafika
- `StatsPlots.jl`: Statistička vizualizacija

Napredni primjer: Metapopulacijski model s genetskom strukturom

Listing 2.12: Julia implementacija genetski strukturiranog metapopulacijskog modela

```
using DifferentialEquations, Plots, LinearAlgebra, Random, Statistics
using DataFrames, CSV, StatsBase

# Definicija strukture za metapopulacijski model
struct GeneticMetapopulationModel{T}
n_patches::Int64 # broj patch-eva
n_alleles::Int64 # broj alela po lokusu
connectivity_matrix::Matrix{T} # matrica povezanosti
carrying_capacity::Vector{T} # nosivost svakog patch-a
migration_rates::Matrix{T} # matrica migracije
selection_coefficients::Vector{T} # selekcijski koeficijenti
mutation_rate::T # stopa mutacije
genetic_drift_strength::T # čjaina genetskog drifta
end

function create_metapopulation_model(;
n_patches::Int = 10,
n_alleles::Int = 2,
landscape_size::Float64 = 100.0,
dispersal_kernel::Float64 = 10.0,
carrying_capacity_mean::Float64 = 1000.0,
carrying_capacity_cv::Float64 = 0.3,
selection_strength::Float64 = 0.01,
mutation_rate::Float64 = 1e-6,
genetic_drift_strength::Float64 = 1.0
)
"""
Kreiranje prostorno-eksplicitnog genetski strukturiranog metapopulacijskog
    modela
"""

# Generiranje čnasuminih pozicija patch-eva
Random.seed!(42)
patch_positions = rand(n_patches, 2) .* landscape_size
```

```
# čRaunanje matrice udaljenosti
distance_matrix = zeros(n_patches, n_patches)
for i in 1:n_patches
for j in 1:n_patches
if i != j
distance_matrix[i, j] = sqrt(sum((patch_positions[i, :] .- patch_positions[j,
     :]).^2))
end
end
end

# Matrica povezanosti (eksponencijalno opadanje s šćudaljenou)
connectivity_matrix = exp.(-distance_matrix ./ dispersal_kernel)
connectivity_matrix[diagind(connectivity_matrix)] .= 0.0

# Normalizacija po redovima
for i in 1:n_patches
row_sum = sum(connectivity_matrix[i, :])
if row_sum > 0
connectivity_matrix[i, :] ./= row_sum
end
end

# Nosivost patch-eva (log-normalna distribucija)
carrying_capacity = exp.(randn(n_patches) .* carrying_capacity_cv .+ log(
    carrying_capacity_mean))

# Matrica migracije (proporcionalna povezanosti)
base_migration_rate = 0.1
migration_rates = connectivity_matrix .* base_migration_rate

# Selekcijski koeficijenti za svaki alel
selection_coefficients = randn(n_alleles) .* selection_strength
selection_coefficients[1] = 0.0 # referentni alel

return GeneticMetapopulationModel(
n_patches,
n_alleles,
connectivity_matrix,
carrying_capacity,
migration_rates,
selection_coefficients,
mutation_rate,
genetic_drift_strength
)
end

function metapopulation_dynamics!(du, u, p, t)
"""
Sustav diferencijalnih žjednadbi za genetski strukturiranu metapopulaciju

State vektor organiziran kao:
[N,  p,  p, ...,  p, N,  p,  p, ...,  p, ...]
gdje je  N ukupna populacija u patch-u i, a  p frekvencija alela j u patch-u
    i
"""
```

```
model = p
n_patches = model.n_patches
n_alleles = model.n_alleles
state_per_patch = n_alleles + 1 # N + frekvencije alela

# Inicijalizacija derivativa
fill!(du, 0.0)

for i in 1:n_patches
# Indeksi za trenutni patch
start_idx = (i-1) * state_per_patch + 1
N_idx = start_idx
allele_start = start_idx + 1
allele_end = start_idx + n_alleles

# Trenutno stanje
N_i = max(u[N_idx], 0.0)
allele_freqs = u[allele_start:allele_end]

# Normalizacija frekvencija alela
total_freq = sum(allele_freqs)
if total_freq > 0
allele_freqs ./= total_freq
else
allele_freqs .= 1.0 / n_alleles
end

# Populacijska dinamika s člogistikim rastom
carrying_capacity = model.carrying_capacity[i]

# Średnja prilagodba (weighted average fitness)
mean_fitness = sum(allele_freqs .* (1.0 .+ model.selection_coefficients))

# Populacijski rast
intrinsic_growth = 0.1 * mean_fitness
dN_local = intrinsic_growth * N_i * (1 - N_i / carrying_capacity)

# Migracija populacije
dN_migration = 0.0
for j in 1:n_patches
if i != j
other_N_idx = (j-1) * state_per_patch + 1
N_j = max(u[other_N_idx], 0.0)

# Imigracija iz patch-a j u patch i
immigration = model.migration_rates[j, i] * N_j

# Emigracija iz patch-a i u patch j
emigration = model.migration_rates[i, j] * N_i

dN_migration += immigration - emigration
end
end

du[N_idx] = dN_local + dN_migration
```

```
        # Dinamika frekvencija alela
        for a in 1:n_alleles
        allele_idx = allele_start + a - 1
        current_freq = allele_freqs[a]

        # Selekcija
        fitness_a = 1.0 + model.selection_coefficients[a]
        dfreq_selection = current_freq * (fitness_a - mean_fitness)

        # Mutacija
        mutation_in = model.mutation_rate * (1.0 - current_freq) / (n_alleles - 1)
        mutation_out = model.mutation_rate * current_freq
        dfreq_mutation = mutation_in - mutation_out

        # Genetski drift (čstohastika komponenta)
        if N_i > 0
        drift_variance = current_freq * (1 - current_freq) / (2 * N_i)
        drift_strength = model.genetic_drift_strength
        dfreq_drift = sqrt(drift_variance) * drift_strength * randn()
        else
        dfreq_drift = 0.0
        end

        # Migracija alela
        dfreq_migration = 0.0
        for j in 1:n_patches
        if i != j
        other_allele_idx = (j-1) * state_per_patch + 1 + a
        other_freq = u[other_allele_idx]
        other_N_idx = (j-1) * state_per_patch + 1
        N_j = max(u[other_N_idx], 0.0)

        if N_i > 0 && N_j > 0
        # Gene flow weighted by migration
        migration_weight = model.migration_rates[j, i]
        dfreq_migration += migration_weight * (other_freq - current_freq)
        end
        end
        end

        du[allele_idx] = dfreq_selection + dfreq_mutation + dfreq_drift +
            dfreq_migration
        end
        end

        return nothing
        end

        function run_genetic_metapopulation_simulation(;
        model_params = Dict(),
        simulation_time = 1000.0,
        n_timepoints = 1000,
        initial_population_fraction = 0.5,
        initial_allele_diversity = 0.1
        )
        """
        Pokretanje simulacije genetski strukturirane metapopulacije
```

```
    """

    println("Kreiranje metapopulacijskog modela...")
    model = create_metapopulation_model(; model_params...)

    n_patches = model.n_patches
    n_alleles = model.n_alleles
    state_per_patch = n_alleles + 1
    total_states = n_patches * state_per_patch

    # čPoetni uvjeti
    println("Postavljanje čpoetnih uvjeta...")
    u0 = zeros(total_states)

    for i in 1:n_patches
    start_idx = (i-1) * state_per_patch + 1
    N_idx = start_idx
    allele_start = start_idx + 1

    # čPoetna populacija
    u0[N_idx] = model.carrying_capacity[i] * initial_population_fraction

    # čPoetne frekvencije alela (blago je varijabilne)
    base_freq = 1.0 / n_alleles
    for a in 1:n_alleles
    allele_idx = allele_start + a - 1
    variation = (rand() - 0.5) * initial_allele_diversity
    u0[allele_idx] = max(0.01, base_freq + variation)
    end

    # Normalizacija frekvencija
    allele_sum = sum(u0[allele_start:allele_start + n_alleles - 1])
    u0[allele_start:allele_start + n_alleles - 1] ./= allele_sum
    end

    # Definicija vremenskog raspona
    tspan = (0.0, simulation_time)
    t_eval = range(0.0, simulation_time, length=n_timepoints)

    # Problem setup
    println("Postavljanje i šrjeavanje diferencijalnih žjednadbi...")
    prob = ODEProblem(metapopulation_dynamics!, u0, tspan, model)

    # šRjeavanje s adaptive time stepping
    sol = solve(prob, Tsit5(), saveat=t_eval, reltol=1e-6, abstol=1e-8)

    return model, sol
    end

    function analyze_genetic_metapopulation_results(model, solution)
    """
    Analiza rezultata genetski strukturirane metapopulacije
    """

    n_patches = model.n_patches
    n_alleles = model.n_alleles
    state_per_patch = n_alleles + 1
```

```
times = solution.t

results = Dict()

# čIzvlaenje vremenskih serija
total_populations = zeros(length(times), n_patches)
allele_frequencies = zeros(length(times), n_patches, n_alleles)

for (t_idx, t) in enumerate(times)
state = solution.u[t_idx]

for i in 1:n_patches
start_idx = (i-1) * state_per_patch + 1
N_idx = start_idx
allele_start = start_idx + 1
allele_end = start_idx + n_alleles

total_populations[t_idx, i] = max(state[N_idx], 0.0)
allele_freqs = state[allele_start:allele_end]

# Normalizacija
total_freq = sum(allele_freqs)
if total_freq > 0
allele_frequencies[t_idx, i, :] = allele_freqs ./ total_freq
else
allele_frequencies[t_idx, i, :] .= 1.0 / n_alleles
end
end
end

# Ukupna metapopulacija
total_metapopulation = sum(total_populations, dims=2)[:, 1]

# Globalne frekvencije alela (weighted by population size)
global_allele_freqs = zeros(length(times), n_alleles)

for t_idx in 1:length(times)
total_pop = total_metapopulation[t_idx]
if total_pop > 0
for a in 1:n_alleles
weighted_freq = sum(total_populations[t_idx, :] .* allele_frequencies[t_idx,
    :, a])
global_allele_freqs[t_idx, a] = weighted_freq / total_pop
end
else
global_allele_freqs[t_idx, :] .= 1.0 / n_alleles
end
end

# Heterozigotnost i Fst analize
expected_heterozygosity = zeros(length(times))
observed_heterozygosity = zeros(length(times))
fst_values = zeros(length(times))

for t_idx in 1:length(times)
# Expected heterozygosity (Hardy-Weinberg)
global_freqs = global_allele_freqs[t_idx, :]
```

```
        He = 1 - sum(global_freqs.^2)
        expected_heterozygosity[t_idx] = He

        # Observed heterozygosity and Fst (simplified calculation)
        patch_He = zeros(n_patches)
        patch_weights = zeros(n_patches)

        for i in 1:n_patches
        if total_populations[t_idx, i] > 0
        local_freqs = allele_frequencies[t_idx, i, :]
        patch_He[i] = 1 - sum(local_freqs.^2)
        patch_weights[i] = total_populations[t_idx, i]
        end
        end

        total_weight = sum(patch_weights)
        if total_weight > 0
        patch_weights ./= total_weight
        Hs = sum(patch_weights .* patch_He) # average within-patch diversity
        observed_heterozygosity[t_idx] = Hs

        # Fst = (Ht - Hs) / Ht
        if He > 0
        fst_values[t_idx] = (He - Hs) / He
        end
        end
        end

        # Populacijska stabilnost
        cv_populations = std(total_populations, dims=1)[:, 1] ./ mean(
            total_populations, dims=1)[:, 1]

        results["times"] = times
        results["total_populations"] = total_populations
        results["total_metapopulation"] = total_metapopulation
        results["allele_frequencies"] = allele_frequencies
        results["global_allele_frequencies"] = global_allele_freqs
        results["expected_heterozygosity"] = expected_heterozygosity
        results["observed_heterozygosity"] = observed_heterozygosity
        results["fst"] = fst_values
        results["population_cv"] = cv_populations

        return results
        end

        function visualize_genetic_metapopulation(model, results)
        """
        Vizualizacija rezultata genetski strukturirane metapopulacije
        """

        times = results["times"]

        # Layout za subplot
        l = @layout [a b c; d e f]

        # Plot 1: Ukupna metapopulacija
        p1 = plot(times, results["total_metapopulation"],
```

```
    linewidth=2, color=:blue,
    xlabel="Vrijeme", ylabel="Ukupna populacija",
    title="Dinamika metapopulacije", grid=true)

    # Plot 2: Populacije po patch-evima
    p2 = plot(xlabel="Vrijeme", ylabel="Populacija po patch-u",
    title="Lokalne populacije", grid=true)

    n_patches_to_show = min(5, model.n_patches)
    for i in 1:n_patches_to_show
    plot!(p2, times, results["total_populations"][:, i],
    linewidth=1, alpha=0.7, label="Patch $i")
    end

    # Plot 3: Globalne frekvencije alela
    p3 = plot(xlabel="Vrijeme", ylabel="Frekvencija alela",
    title="Globalna evolucija alela", grid=true)

    colors = [:red, :green, :blue, :orange, :purple]
    for a in 1:model.n_alleles
    color = colors[mod(a-1, length(colors)) + 1]
    plot!(p3, times, results["global_allele_frequencies"][:, a],
    linewidth=2, color=color, label="Alel $a")
    end

    # Plot 4: Genetska raznolikost
    p4 = plot(times, results["expected_heterozygosity"],
    linewidth=2, color=:red, label="Expected (Ht)",
    xlabel="Vrijeme", ylabel="Heterozigotnost",
    title="Genetska raznolikost", grid=true)
    plot!(p4, times, results["observed_heterozygosity"],
    linewidth=2, color=:blue, label="Observed (Hs)")

    # Plot 5: Fst kroz vrijeme
    p5 = plot(times, results["fst"],
    linewidth=2, color=:purple,
    xlabel="Vrijeme", ylabel="Fst",
    title="Genetska diferencijacija", grid=true)

    # Plot 6: Stabilnost populacija
    p6 = bar(1:model.n_patches, results["population_cv"],
    xlabel="Patch ID", ylabel="Koeficijent varijacije",
    title="Stabilnost lokalnih populacija",
    color=:lightblue, alpha=0.7)

    # Kombinacija svih plotova
    final_plot = plot(p1, p2, p3, p4, p5, p6, layout=l, size=(1200, 800))

    return final_plot
    end

    # Pokretanje primjera
    function main_genetic_metapopulation_example()
    """
    Glavni primjer pokretanja genetski strukturirane metapopulacije
    """
```

```
println("=== GENETSKI STRUKTURIRANA METAPOPULACIJA ===")

# Parametri simulacije
model_params = Dict(
:n_patches => 8,
:n_alleles => 3,
:landscape_size => 50.0,
:dispersal_kernel => 8.0,
:carrying_capacity_mean => 800.0,
:carrying_capacity_cv => 0.4,
:selection_strength => 0.02,
:mutation_rate => 1e-5,
:genetic_drift_strength => 0.5
)

# Pokretanje simulacije
model, solution = run_genetic_metapopulation_simulation(
model_params = model_params,
simulation_time = 500.0,
n_timepoints = 500,
initial_population_fraction = 0.6,
initial_allele_diversity = 0.2
)

# Analiza rezultata
println("Analiza rezultata...")
results = analyze_genetic_metapopulation_results(model, solution)

# Ispis čkljunih rezultata
println("\n=== ČKLJUNI REZULTATI ===")
println("čKonana ukupna populacija: $(round(Int, results["
    total_metapopulation"][end]))")

final_global_freqs = results["global_allele_frequencies"][end, :]
for (i, freq) in enumerate(final_global_freqs)
println("čKonana frekvencija alela $i: $(round(freq, digits=4))")
end

final_fst = results["fst"][end]
println("čKonani Fst: $(round(final_fst, digits=4))")

final_heterozygosity = results["expected_heterozygosity"][end]
println("čKonana heterozigotnost: $(round(final_heterozygosity, digits=4))")

# Vizualizacija
println("Kreiranje vizualizacija...")
final_plot = visualize_genetic_metapopulation(model, results)
display(final_plot)

return model, solution, results, final_plot
end

# Pozivanje glavnog primjera
# main_genetic_metapopulation_example()
```

Instalacija i upravljanje paketima

Listing 2.13: Instalacija Julia

Instalacija Julia

```
# Download binaries from https://julialang.org/downloads/

# Linux - using juliaup (recommended)
curl -fsSL https://install.julialang.org | sh

# MacOS - using Homebrew
brew install julia

# Alternativno, direktno preuzimanje
wget https://julialang-s3.julialang.org/bin/linux/x64/1.9/julia-1.9.3-linux-
    x86_64.tar.gz
tar -xzf julia-1.9.3-linux-x86_64.tar.gz
sudo mv julia-1.9.3 /opt/
sudo ln -s /opt/julia-1.9.3/bin/julia /usr/local/bin/julia
```

Listing 2.14: Julia package management

Upravljanje paketima u Julia

```
# Ulazak u Pkg REPL mod (pritisnuti ']' u Julia REPL)
]

# Dodavanje paketa
add DifferentialEquations
add Plots DataFrames CSV

# Dodavanje čspecifine verzije
add DifferentialEquations@6.15

# Dodavanje iz GitHub repozitorija
add https://github.com/user/PackageName.jl

# žAuriranje paketa
update

# Status instaliranih paketa
status

# Uklanjanje paketa
remove PackageName

# Kreiranje projektnog žokruenja
activate .
instantiate # instaliraj sve pakete iz Project.toml

# Testiranje paketa
test DifferentialEquations

# Izlazak iz Pkg moda (pritisnuti backspace)
```

Listing 2.15: Project.toml za ekološki projekt

Project.toml za reproducibilnost

```
name = "EcologicalModeling"
version = "0.1.0"

[deps]
DifferentialEquations = "0c46a032-eb83-5123-abaf-570d42b7fbaa"
Plots = "91a5bcdd-55d7-5caf-9e0b-520d859cae80"
DataFrames = "a93c6f00-e57d-5684-b7b6-d8193f3e46c0"
CSV = "336ed68f-0bac-5ca0-87d4-7b16caf5d00b"
StatsBase = "2913bbd2-ae8a-5f71-8c99-4fb6c76f3a91"
Distributions = "31c24e10-a181-5473-b8eb-7969acd0382f"
LinearAlgebra = "37e2e46d-f89d-539d-b4ee-838fcccc9c8e"
Random = "9a3f8284-a2c9-5f02-9a11-845980a1fd5c"
Statistics = "10745b16-79ce-11e8-11f9-7d13ad32a3b2"

[compat]
julia = "1.8"
DifferentialEquations = "7"
Plots = "1.35"
DataFrames = "1.4"
```

2.4.8 NetLogo

Uvod u NetLogo

NetLogo je programsko okruženje specijalno dizajnirano za agent-based modeliranje (ABM). Razvio ga je Uri Wilensky na Northwestern University. NetLogo omogućava modeliranje složenih sustava koji nastaju iz interakcija mnoštva individualnih agenata, što ga čini idealnim za ekološko modeliranje na razini jedinki i populacija.

Ključne značajke NetLogo-a

- **Agent-based pristup**: Modeliranje individualnih organizama
- **Prostorna eksplicitnost**: Ugrađena 2D mrežna struktura
- **Vizualna priroda**: Real-time vizualizacija simulacija
- **Jednostavnost**: Relativno lako učenje
- **Biblioteca modela**: Opsežna kolekcija gotovih modela

Struktura NetLogo modela

NetLogo model se sastoji od četiri glavna dijela:

1. **Interface**: Korisničko sučelje s kontrolama i grafikonima
2. **Code**: Programski kod modela
3. **Info**: Dokumentacija modela
4. **Procedures**: Funkcije i procedure

Primjer NetLogo koda: Predator-Prey model

Listing 2.16: Agent-based predator-prey model

```
; Globalne varijable
```

```
globals [
grass-regrowth-time ; vrijeme potrebno za ponovni rast trave
sheep-gain-from-food ; energija koju ovca dobiva od trave
wolf-gain-from-food ; energija koju vuk dobiva od ovce
sheep-reproduce ; vjerojatnost reprodukcije ovaca
wolf-reproduce ; vjerojatnost reprodukcije vukova

; Varijable za ćpraenje populacija
initial-number-sheep
initial-number-wolves
]

; Definiranje tipova agenata
breed [sheep a-sheep]
breed [wolves wolf]

; Svojstva patch-eva ć(elija žmree)
patches-own [
countdown ; vrijeme do ponovnog rasta trave
]

; Svojstva ovaca
sheep-own [
energy ; energija ovce
]

; Svojstva vukova
wolves-own [
energy ; energija vuka
]

; Procedura za pokretanje modela
to setup
clear-all

; Postavljanje parametara
set grass-regrowth-time 30
set sheep-gain-from-food 4
set wolf-gain-from-food 20
set sheep-reproduce 4
set wolf-reproduce 5

; Kreiranje čpoetne populacije ovaca
create-sheep initial-number-sheep [
setxy random-xcor random-ycor
set shape "sheep"
set color white
set size 1.5
set energy random (2 * sheep-gain-from-food)
]

; Kreiranje čpoetne populacije vukova
create-wolves initial-number-wolves [
setxy random-xcor random-ycor
set shape "wolf"
set color black
set size 2
```

```
set energy random (2 * wolf-gain-from-food)
]

; Inicijalizacija trave
ask patches [
set pcolor green
set countdown grass-regrowth-time
]

reset-ticks
end
```

2.4.9 Reproducibilnost i najbolje prakse

Principi reproducibilne znanosti

Reproducibilnost je temeljan zahtjev znanstvenog rada. U ekološkom modeliranju, reproducibilnost znači da treći ljudi mogu ponoviti analize i dobiti iste rezultate koristeći iste podatke i metode.

Ključni elementi reproducibilnosti:

- **Verzioniranje koda**: Korištenje Git-a ili sličnih sustava
- **Dokumentacija**: Jasna dokumentacija svih koraka
- **Upravljanje ovisnostima**: Specificiranje verzija svih paketa
- **Kontejnerizacija**: Docker, Singularity za potpunu reproducibilnost
- **Dijeljenje podataka**: Otvoreni podaci u standardnim formatima

Git verzioniranje za ekološke projekte

Listing 2.17: Git workflow za ekološke projekte

```
# Inicijalizacija repozitorija
git init ecology_project
cd ecology_project

# Kreiranje .gitignore datoteke
echo "*.Rdata
*.Rhistory
.RData
.Ruserdata
__pycache__/
*.pyc
*.pyo
data/raw/
results/temp/
.ipynb_checkpoints/" > .gitignore

# Struktura direktorija
mkdir -p {data/{raw,processed},scripts,results,docs,figures}

# Prvi commit
git add .
```

```
git commit -m "Initial project structure"

# Kreiranje brancha za novu analizu
git checkout -b feature/population-analysis

# Dodavanje remote repozitorija
git remote add origin https://github.com/username/ecology_project.git

# Push to remote
git push -u origin main
```

Struktura reproducibilnog projekta

Listing 2.18: Preporučena struktura direktorija

```
ecology_project/
 README.md # Opis projekta i upute
 LICENSE # Licenca
 environment.yml # Conda environment (Python)
 renv.lock # R environment
 requirements.txt # Python paketi
 Dockerfile # Container definicija
 Makefile # Automatizacija workflow-a
 .gitignore # Git ignore datoteka
 data/
    raw/ # Sirovi podaci (read-only)
    processed/ # đObraeni podaci
    external/ # Eksterni podaci
 scripts/
    01_data_preparation.R
    02_exploratory_analysis.py
    03_modeling.R
    04_visualization.py
 src/ # Izvorni kod funkcija
    R/
    python/
 results/
    figures/
    tables/
    models/
 docs/
    methodology.md
    analysis_notes.md
 tests/ # Unit testovi
 test_functions.R
 test_models.py
```

Docker kontejneri za ekološko modeliranje

Listing 2.19: Dockerfile za R + Python okruženje

```
# Bazna slika s R-om
FROM rocker/verse:4.3.0

# Metapodaci
```

```
LABEL maintainer="your.email@institution.org"
LABEL description="Reproducible environment for ecological modeling"

# Systemski paketi
RUN apt-get update && apt-get install -y \
python3 \
python3-pip \
python3-venv \
gdal-bin \
libgdal-dev \
libgeos-dev \
libproj-dev \
&& rm -rf /var/lib/apt/lists/*

# R paketi
RUN install2.r --error \
dismo \
vegan \
mgcv \
lme4 \
sf \
raster \
ggplot2 \
dplyr \
tidyr \
knitr \
rmarkdown

# Python žokruenje
RUN python3 -m venv /opt/venv
ENV PATH="/opt/venv/bin:$PATH"

# Python paketi
COPY requirements.txt /tmp/
RUN pip install --no-cache-dir -r /tmp/requirements.txt

# Kopiranje projekta
WORKDIR /home/rstudio
COPY . .

# Postavljanje dozvola
RUN chown -R rstudio:rstudio /home/rstudio

# Default naredba
CMD ["R"]
```

Automatizacija s Makefile

Listing 2.20: Makefile za automatizaciju analize

```
# Makefile za šekoloki modelski projekt

# Varijable
R_SCRIPTS = scripts
PYTHON_SCRIPTS = scripts
DATA_DIR = data
```

```
RESULTS_DIR = results
FIGURES_DIR = $(RESULTS_DIR)/figures

# Glavni cilj
all: data analysis figures report

# Priprema podataka
data: $(DATA_DIR)/processed/cleaned_data.csv

$(DATA_DIR)/processed/cleaned_data.csv: $(DATA_DIR)/raw/field_data.csv
Rscript $(R_SCRIPTS)/01_data_preparation.R

# Eksploracijska analiza
analysis: $(RESULTS_DIR)/exploratory_analysis.html

$(RESULTS_DIR)/exploratory_analysis.html: $(DATA_DIR)/processed/cleaned_data.
    csv
python $(PYTHON_SCRIPTS)/02_exploratory_analysis.py

# Modeliranje
models: $(RESULTS_DIR)/models/population_model.rds

$(RESULTS_DIR)/models/population_model.rds: $(DATA_DIR)/processed/
    cleaned_data.csv
Rscript $(R_SCRIPTS)/03_modeling.R

# Figure
figures: $(FIGURES_DIR)/population_dynamics.png

$(FIGURES_DIR)/population_dynamics.png: $(RESULTS_DIR)/models/
    population_model.rds
python $(PYTHON_SCRIPTS)/04_visualization.py

# Finalni izvještaj
report: $(RESULTS_DIR)/final_report.pdf

$(RESULTS_DIR)/final_report.pdf: $(FIGURES_DIR)/population_dynamics.png
Rscript -e "rmarkdown::render('docs/report.Rmd', output_dir='$(RESULTS_DIR)
    ')"

# Čišćenje
clean:
rm -rf $(RESULTS_DIR)/*
rm -rf $(DATA_DIR)/processed/*

# Testiranje
test:
Rscript tests/test_functions.R
python -m pytest tests/

# Docker build
docker-build:
docker build -t ecology-project .

# Docker run
docker-run:
docker run -it --rm -v $(PWD):/home/rstudio ecology-project
```

```
.PHONY: all data analysis models figures report clean test docker-build
    docker-run
```

Upravljanje ovisnostima

R renv sustav

Listing 2.21: Upravljanje R ovisnostima s renv

```
# Inicijalizacija renv projekta
renv::init()

# Instalacija paketa
install.packages("dismo")
renv::snapshot()

# žAuriranje lockfile-a nakon promjena
renv::snapshot()

# ćVraanje na verzije iz lockfile-a
renv::restore()

# žAuriranje svih paketa
renv::update()

# Status trenutnog žokruenja
renv::status()

# Kreiranje portable library
renv::isolate()
```

Python virtualenv i pip-tools

Listing 2.22: Python dependency management

```
# Kreiranje virtualnog žokruenja
python -m venv ecology_env
source ecology_env/bin/activate

# Instalacija pip-tools
pip install pip-tools

# Kreiranje requirements.in
echo "numpy>=1.20.0
scipy>=1.7.0
pandas>=1.3.0
matplotlib>=3.4.0
scikit-learn>=1.0.0" > requirements.in

# Generiranje lockovanog requirements.txt
pip-compile requirements.in

# Instalacija čtonih verzija
pip-sync requirements.txt

# žAuriranje ovisnosti
pip-compile --upgrade requirements.in
```

2.4.10 Integracijska prilazi i interoperabilnost

R i Python integracija

reticulate paket

Listing 2.23: Pozivanje Python-a iz R-a

```
library(reticulate)

# Specificiranje Python žokruenja
use_virtualenv("ecology_env")

# Import Python modula
np <- import("numpy")
plt <- import("matplotlib.pyplot")

# šKoritenje Python funkcija
data <- np$random$normal(0, 1, 1000L)
hist(py_to_r(data))

# šIzvravanje Python koda
py_run_string("
import numpy as np
result = np.mean([1, 2, 3, 4, 5])
")

# Pristup Python varijablama
py$result
```

2.4.11 Zaključak i preporuke

Izbor alata prema vrsti problema

Tablica 2.1: Preporučeni alati prema vrsti modeliranja

Vrsta modeliranja	Primarni alat	Alternativni alati	Specijali
Statistička analiza	R	Python (statsmodels)	Julia (Sta
Prostorno modeliranje	R (sf, raster)	Python (geopandas)	Julia (Ge
Strojno učenje	Python	R (caret, tidymodels)	Julia (ML
Agent-based modeli	NetLogo	Python (Mesa), R	Julia (Ag
Kompleksni ODE sustavi	Julia (DiffEq.jl)	MATLAB, Python (scipy)	GNU Oct
Stohastičke diferencijalne jednadžbe	Julia (StochasticDiffEq.jl)	MATLAB	Python (s
Bayesovska analiza	R (Stan)	Python (PyMC)	Julia (Tu
Numerička optimizacija	Julia (Optim.jl)	MATLAB (Optimization)	GNU Oct
Visokomperformantno računanje	Julia	Rust, C/C++, Fortran	Python (N
Paralelno računanje	Julia	Python (multiprocessing)	MATLAE
MATLAB kod migracija	GNU Octave	Julia	Python (N
Vizualizacija	R (ggplot2)	Python (matplotlib, plotly)	Julia (Plo
Interaktivna analiza	Python (Jupyter)	R (RMarkdown)	Julia (Plu
Reproducibilnost	R (renv)	Python (conda/pip)	Julia (Pk
Genetička algoritmi	Python (DEAP)	R (GA)	Julia (Ev
Mrežna analiza	R (igraph)	Python (NetworkX)	Julia (Gra

Nastavlja se n

Tablica 2.1 – nastavak s prethodne stranice

Vrsta modeliranja	Primarni alat	Alternativni alati	Specijali
Vremenske serije	R (forecast)	Python (statsmodels)	Julia (Tir
Prostorna statistika	R (gstat, sp)	Python (geostatspy)	Julia (Ge
Monte Carlo simulacije	Julia	Python, R	MATLAE
Simboličko računanje	Python (SymPy)	MATLAB (Symbolic)	Julia (Sy

Objašnjenja i smjernice za izbor

GNU Octave specijalizacije:

- **MATLAB migracija**: 95%+ sintaksna kompatibilnost omogućava direktno pokretanje postojećeg MATLAB koda
- **Obrazovne institucije**: Besplatan pristup MATLAB-ovalnim mogućnostima
- **Numerička stabilnost**: Koristi iste BLAS/LAPACK biblioteke kao MATLAB
- **Ograničenja**: Sporiji od MATLAB-a, manje toolbox-ova, slabija 3D grafika

Julia specijalizacije:

- **Diferencijalne jednadžbe**: DifferentialEquations.jl najnapredniji je ekosustav za DE
- **Visokomperformantno računanje**: "Two language problem" rješavanje - brzina C-a s jednostavnošću Python-a
- **Paralelizam**: Ugrađena podrška za distribuirano računanje i GPU
- **Znanstvena područja**: Posebno snažan u fizici, biologiji, ekonomiji
- **Ograničenja**: Mlađa zajednica, manji broj paketa od Python-a/R-a

Hibridni pristupi:

- **R + Julia**: R za statistiku i vizualizaciju, Julia za računalno zahtjevne simulacije
- **Python + Julia**: Python za ML i podatkovnu analizu, Julia za numeričko modeliranje
- **MATLAB/Octave + R**: MATLAB/Octave za inženjerske simulacije, R za statističku analizu
- **NetLogo + R/Python**: NetLogo za ABM prototipiranje, R/Python za skalabilne implementacije

Ključne preporuke

1. **Kombiniraj alate**: Koristi prednosti svakog alata

   Nijedan alat nije savršen za sve situacije. R je izvrsno za statistiku i vizualizaciju, Python za strojno učenje i skalabilnost, MATLAB za numeričke simulacije, NetLogo za agent-based modele. Kombiniraj ih strategijski - koristi R za početnu analizu i čišćenje podataka, prebaci u Python za složene algoritme strojnog učenja, a rezultate vizualiziraj u R-u s ggplot2. Takav hibridni pristup omogućava maksimalnu iskoristivost specifičnih prednosti svakog alata.

2. **Dokumentiraj sve**: Kod, podatke, pretpostavke, ograničenja

   Dobra dokumentacija je temelj reproducibilne znanosti. Svaki kod trebao bi imati jasne komentare koji objašnjavaju logiku, svaki dataset opise varijabli i metoda prikupljanja, svaki model eksplicitne pretpostavke. Kreiraj README datoteke za projekte, koristi comment tagove u kodu, dokumentiraj sve odluke o čišćenju podataka. Dokumentiraj također ograničenja modela, nesigurnosti u podacima i pretpostavke koje utječu na interpretaciju rezultata.

3. **Testiraj redovito**: Unit testovi, validacija modela

   Sistematsko testiranje sprječava greške i povećava pouzdanost rezultata. Piši unit testove koji provjeravaju da funkcije rade kako je namijenjeno, testiraj modele na poznatim dataset s poznatim rezultatima, provjeravaj granične slučajeve. U R-u koristi pakete testthat ili tinytest, u Python-u pytest. Također validiraj model performanse kroz cross-validation, bootstrap analize i osjetljivost analize.

4. **Verzitrajte promjene**: Git za kod, DVC za podatke

   Verzianje omogućava praćenje promjena i vraćanje na prethodne verzije ako nešto pođe po zlu. Git je standard za verzioniranje koda- *"Commitajte"* često s opisnim porukama, koristite *"branching"* za eksperimente, *"maintainajte"* čist glavni *"branch"*. Za velike skupove podataka neprikladne za Git, koristite DVC (*Data Version Control*) kako bi pratili promjene podataka i omogućili reproduciranje točnih verzija podataka za svaki eksperiment.

5. **Dijelite otvoreno**: Kod, podaci, rezultati

   Otvorena znanost povećava transparentnost i omogućava drugima da grade na vašem radu. Objavljujte kod na GitHub/GitLab s jasnim licencama, dijelite skupove podataka na repozitorijima poput Dryad ili Zenodo, napravite preprinte dostupnima na bioRxiv. Koristite FAIR princip (*Findable, Accessible, Interoperable, Reusable*) za sve digitalne objekte jer tako ne samo što pomažete znanosti već često rezultira boljim citiranjem vaših radova.

6. **Planirajte skalabilnost**: Optimizacija za veće probleme

   Dizajnirajte kod koji može rasti s vašim potrebama. Koristite vektorske operacije umjesto petlji, leveraging paralelizaciju za simulacije, razmislite o *memory-efficient algoritmima* za velike skupove podataka. Ako trenutno radite s 1000 podatkovnih točaka, zapitajte se sebe hoćete li kod raditi s milijun točaka? Koristite *profiling* alate za identificiranje *bottleneck-a* i optimizirajte kritične dijelove koda. *Cloud computing* platfome mogu omogućiti skaliranje izvan lokalnih ograničenja.

7. **Radite sigurnosne kopije**: Redundantno pohranjivanje

   Gubitak podataka i koda može uništiti mjesece rada. Implementirajte "3-2-1" pravilo: 3 kopije (izvornik + 2 sigurnosne), na 2 različita media (lokalni disk + cloud), s 1 kopijom geografski odvojenom. Automatskim *cloud sync* za aktivne projekte, periodični *backup* za sve skupova epodataka, *offsite storage* za kritične dugotrajne projekt. Git repozitoriji na GitHub/GitLab automatski pružaju *distributed backup* za kod.

8. **Učici kontinuirano**: Novi alati, tehnike, najbolje prakse

   Ekološko modeliranje je dinamično polje s konstantnim novostima. Prati znanstvene journal, blog (R-bloggers, Towards Data Science), attend konference i webinar. Eksperimentiraj s novim paketima i tehnologiji na malim projekt prije implementacije u važan rad. Networking s drugim ekolozim-modelerima kroz Twitter, ResearchGate ili

profesionalna udruženja omogućava razmjenu znanja i kolaboracije. Izdvoj vrijeme za redovno osvježavanje vještina.

Dio II

Populacije, zajednice i ekosustavi

Poglavlje 3

Modeli populacijske dinamike

3.1 Jednostavni modeli

3.1.1 Uvod u populacijsku dinamiku

Populacijska dinamika proučava promjene u veličini i strukturi populacija kroz vrijeme. Temelji se na osnovnim demografskim procesima: rađanju, smrti, imigraciji i emigraciji. Matematičko modeliranje populacijske dinamike omogućava razumijevanje faktora koji utječu na rast ili opadanje populacija te predviđanje budućih trendova.

Opća bilancijska jednadžba za populaciju može se zapisati kao:

$$\frac{dN}{dt} = \text{Rođenja} - \text{Smrti} + \text{Imigracija} - \text{Emigracija} \tag{3.1}$$

3.1.2 Eksponencijalni rast

Kontinuirani model eksponencijalnog rasta

Najjednostavniji model populacijske dinamike pretpostavlja da je stopa rasta populacije proporcionalna trenutnoj veličini populacije:

$$\frac{dN}{dt} = rN \tag{3.2}$$

gdje je:

- $N(t)$ - veličina populacije u vremenu t
- r - intrinsična stopa rasta (per capita)

Rješenje diferencijalne jednadžbe:

$$N(t) = N_0 e^{rt} \tag{3.3}$$

gdje je N_0 početna veličina populacije u $t = 0$.

Interpretacija parametra r

Intrinsična stopa rasta r može se razložiti na komponente:

$$r = b - d + i - e \tag{3.4}$$

gdje su:

- b - stopa rođenja (births per capita)

- d - stopa smrti (deaths per capita)
- i - stopa imigracije
- e - stopa emigracije

Za zatvorenu populaciju (bez migracije): $r = b - d$

Vrijeme udvostručavanja

Vrijeme potrebno da se populacija udvostručuje:

$$t_{\text{double}} = \frac{\ln(2)}{r} \approx \frac{0.693}{r} \tag{3.5}$$

Diskretni eksponencijalni model

Za populacije s diskretnim generacijama:

$$N_{t+1} = \lambda N_t \tag{3.6}$$

gdje je λ konačna stopa rasta (finite rate of increase).
Veza između r i λ:

$$\lambda = e^r \quad \text{ili} \quad r = \ln(\lambda) \tag{3.7}$$

Opće rješenje:

$$N_t = N_0 \lambda^t \tag{3.8}$$

3.1.3 Logistički rast

Kontinuirani logistički model

Eksponencijalni rast ne može se nastaviti beskonačno zbog ograničenosti resursa. Logistički model uključuje koncept nosivosti staništa (K):

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) \tag{3.9}$$

gdje je K nosivost staništa (carrying capacity).

Analiza logističke jednadžbe

Ravnotežne točke:

- $N^* = 0$ (nestabilna)
- $N^* = K$ (stabilna)

Rješenje logističke jednadžbe:
Koristeći separaciju varijabli:

$$N(t) = \frac{K}{1 + \left(\frac{K - N_0}{N_0}\right) e^{-rt}} \tag{3.10}$$

ili ekvivalentno:

$$N(t) = \frac{K N_0 e^{rt}}{K + N_0(e^{rt} - 1)} \tag{3.11}$$

Karakteristike logističke krivulje

Točka infleksije:
Maksimalna stopa rasta postiže se kada je $N = K/2$:

$$\left.\frac{d^2N}{dt^2}\right|_{N=K/2} = 0 \tag{3.12}$$

Maksimalna stopa rasta:

$$\left.\frac{dN}{dt}\right|_{\max} = \frac{rK}{4} \tag{3.13}$$

Asimptotsko ponašanje:

$$\lim_{t\to\infty} N(t) = K \tag{3.14}$$

$$\lim_{t\to-\infty} N(t) = 0 \tag{3.15}$$

3.1.4 Diskretni modeli rasta

Diskretni logistički model

$$N_{t+1} = N_t + rN_t\left(1 - \frac{N_t}{K}\right) \tag{3.16}$$

ili ekvivalentno:

$$N_{t+1} = N_t\left(1 + r\left(1 - \frac{N_t}{K}\right)\right) \tag{3.17}$$

Ravnotežna točka:

$$N^* = K \tag{3.18}$$

Stabilnost: Ovisi o vrijednosti r

- $0 < r < 2$: stabilan pristup ravnoteži
- $r > 2$: oscilacije ili kaos

Ricker model

Ricker model posebno je pogodan za populacije s preklapajućim generacijama i gustoćno ovisnom reprodukcijom:

$$N_{t+1} = N_t e^{r\left(1-\frac{N_t}{K}\right)} \tag{3.19}$$

Linearizacija oko ravnoteže:
Oko $N^* = K$:

$$N_{t+1} - K \approx -r(N_t - K) \tag{3.20}$$

Uvjeti stabilnosti:

- $0 < r < 2$: monotonski pristup
- $2 < r < 2.526$: oscilirajući pristup
- $r > 2.526$: kaotična dinamika

Beverton-Holt model

Model posebno korišten u ribarstvu:

$$N_{t+1} = \frac{aN_t}{1 + bN_t} \tag{3.21}$$

gdje su a i b pozitivni parametri.

Veza s logističkim parametrima:

$$a = e^r \tag{3.22}$$

$$b = \frac{e^r - 1}{K} \tag{3.23}$$

Ravnotežna točka:

$$N^* = \frac{a - 1}{b} \tag{3.24}$$

Stabilnost: Beverton-Holt model uvijek konvergira monotonski prema ravnoteži za $a > 1$.

3.1.5 Usporedba diskretnih modela

Tablica 3.1: Usporedba diskretnih populacijskih modela

Model	**Jednadžba**	**Karakteristike**	**Primjena**
Eksponencijalni	$N_{t+1} = \lambda N_t$	Neograničen rast	Početni rast populacije
Diskretni logistički	$N_{t+1} = N_t(1 + r(1 - N_t/K))$	Može biti nestabilan	Jednostavni modeli
Ricker	$N_{t+1} = N_t e^{r(1-N_t/K)}$	Prekomjerna kompenzacija	Ribe, insekti
Beverton-Holt	$N_{t+1} = \frac{aN_t}{1+bN_t}$	Uvijek stabilan	Ribarstvo

3.1.6 Procjena parametara

Linearizacija za procjenu parametara

Eksponencijalni model:

$$\ln(N_t) = \ln(N_0) + rt \tag{3.25}$$

Linearna regresija $\ln(N_t)$ na t daje procjenu r.

Logistički model:

Gompertz transformacija:

$$\ln\left(\frac{K - N_t}{N_t}\right) = \ln\left(\frac{K - N_0}{N_0}\right) - rt \tag{3.26}$$

Nelinearna regresija

Za direktnu procjenu parametara logističke jednadžbe koristi se nelinearna regresija s funkcijom:

$$N(t) = \frac{K}{1 + ae^{-rt}} \tag{3.27}$$

gdje je $a = (K - N_0)/N_0$.

Početne vrijednosti za optimizaciju:

- K: maksimalna opažena vrijednost $\times 1.2$

- r: gradijent u točki infleksije
- a: iz početnih uvjeta

3.2 Dobno/stanovno strukturirani modeli

3.2.1 Uvod u strukturirane modele

Jednostavni populacijski modeli tretiraju sve jedinke kao identične. U stvarnosti, demografi parametri (rođenja, smrti) uvelike ovise o dobi, veličini ili reproduktivnom stanju jedinki. Strukturirani modeli eksplicitno modeliraju ove razlike.

3.2.2 Leslie matrica

Teorijski okvir

Leslie matrica, koju je razvio P.H. Leslie 1945., model je za dobno strukturirane populacije s diskretnim vremenskim koracima.

Populacija se dijeli na n dobnih klasa. Stanje populacije u vremenu t opisuje vektor:

$$\mathbf{n}_t = \begin{pmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_n(t) \end{pmatrix} \tag{3.28}$$

gdje je $n_i(t)$ broj jedinki u i-toj dobnoj klasi.

Konstrukcija Leslie matrice

Leslie matrica ima oblik:

$$\mathbf{L} = \begin{pmatrix} F_1 & F_2 & F_3 & \cdots & F_{n-1} & F_n \\ S_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & S_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & S_{n-1} & 0 \end{pmatrix} \tag{3.29}$$

gdje su:

- F_i - plodnost i-te dobne klase (broj mladih po jedinki)
- S_i - preživljavanje iz i-te u $(i+1)$-u dobnu klasu

Populacijska projekcija:

$$\mathbf{n}_{t+1} = \mathbf{L}\mathbf{n}_t \tag{3.30}$$

Asimptotsko ponašanje

Za ergodičnu Leslie matricu (sve $F_i, S_i \geq 0$, $F_i > 0$ za neki i):

Dominantna svojstvena vrijednost λ_1:

$$\lambda_1 = \text{najveća svojstvena vrijednost od } \mathbf{L} \tag{3.31}$$

Asimptotska stopa rasta:

$$r = \ln(\lambda_1) \tag{3.32}$$

Stabilna dobna distribucija:

Odgovarajući svojstveni vektor **w**:

$$\mathbf{Lw} = \lambda_1 \mathbf{w} \tag{3.33}$$

Normalizirano: $\sum w_i = 1$
Reproduktivna vrijednost:
Lijevi svojstveni vektor **v**:

$$\mathbf{v}^T\mathbf{L} = \lambda_1 \mathbf{v}^T \tag{3.34}$$

v_i predstavlja budući reproduktivni potencijal jedinke u dobnoj klasi i.

Euler-Lotka jednadžba

Za kontinuirane dobne distribucije, karakteristična jednadžba Leslie matrice postaje:

$$1 = \int_0^\infty e^{-ra} l(a) m(a) da \tag{3.35}$$

gdje su:

- $l(a)$ - vjerojatnost preživljavanja do dobi a
- $m(a)$ - maternalna funkcija (rođenja po jedinki dobi a)
- r - intrinsična stopa rasta

3.2.3 Lefkovitch matrica

Općeniti okvir

Lefkovitch matrica generalizacija je Leslie matrice za populacije strukturirane prema bilo kojoj karakteristici (veličina, stadij razvoja, reproduktivno stanje).

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{3.36}$$

gdje a_{ij} predstavlja doprinos jedinki iz klase j klasi i u sljedećem vremenskom koraku.

Interpretacija elemenata matrice

Dijagonalni elementi (a_{ii}): Ostanak u istoj klasi **Iznad dijagonale:** Napredovanje u veće klase **Ispod dijagonale:** Regresija u manje klase **Prvi red:** Reprodukcija (novi potomci)
Ograničenja:

$$a_{ij} \geq 0 \quad \forall i, j \tag{3.37}$$

$$\sum_{i=2}^{n} a_{ij} \leq 1 \quad \forall j \geq 2 \text{ (bez prvog reda)} \tag{3.38}$$

3.2.4 Analiza osjetljivosti i elastičnosti

Osjetljivost (Sensitivity)

Osjetljivost λ na promjene u elementi matrice a_{ij}:

$$s_{ij} = \frac{\partial \lambda}{\partial a_{ij}} = \frac{v_i w_j}{\langle \mathbf{v}, \mathbf{w} \rangle} \tag{3.39}$$

gdje je $\langle \mathbf{v}, \mathbf{w} \rangle = \sum v_i w_i$ skalarni produkt.

Svojstva osjetljivosti:

- $s_{ij} > 0$ za sve elemente
- $\sum_{i,j} s_{ij} a_{ij} = \lambda$ (sumiranje po svim elementima)

Elastičnost (Elasticity)

Relativan utjecaj proporcionalnih promjena:

$$e_{ij} = \frac{a_{ij}}{\lambda} \frac{\partial \lambda}{\partial a_{ij}} = \frac{a_{ij} s_{ij}}{\lambda} \tag{3.40}$$

Interpretacija: e_{ij} predstavlja proporcionalnu promjenu u λ koja rezultira iz 1% promjene u a_{ij}.

Svojstva elastičnosti:

$$\sum_{i,j} e_{ij} = 1 \tag{3.41}$$

$$0 \leq e_{ij} \leq 1 \tag{3.42}$$

LTRE analiza (Life Table Response Experiment)

LTRE kvantificira razlike u λ između populacija:

$$\lambda^{(1)} - \lambda^{(2)} \approx \sum_{i,j} s_{ij} (a_{ij}^{(1)} - a_{ij}^{(2)}) \tag{3.43}$$

gdje gornji indeksi označavaju različite populacije ili tretmane.

Doprinos elementa a_{ij}:

$$c_{ij} = s_{ij} (a_{ij}^{(1)} - a_{ij}^{(2)}) \tag{3.44}$$

3.2.5 Napredni koncepti strukturiranih modela

Gustoćno ovisni strukturirani modeli

$$\mathbf{n}_{t+1} = \mathbf{A}(N_t) \mathbf{n}_t \tag{3.45}$$

gdje $\mathbf{A}(N_t)$ ovisi o ukupnoj populaciji $N_t = \sum n_i(t)$.

Primjer - logistička regulacija reprodukcije:

$$F_i(N_t) = F_i^{\max} \left(1 - \frac{N_t}{K} \right) \tag{3.46}$$

Stohastički strukturirani modeli

Elementarna stohastičnost:

$$\mathbf{n}_{t+1} = \mathbf{A}_t \mathbf{n}_t \tag{3.47}$$

gdje su elementi $\mathbf{A}_t$ slučajne varijable.

Stohastička stopa rasta:

$$r_s = \lim_{t\to\infty} \frac{1}{t} \ln(N_t) \tag{3.48}$$

Za nekorelirana okruženja:

$$r_s \approx E[\ln(\lambda_t)] - \frac{1}{2}\text{Var}[\ln(\lambda_t)] \tag{3.49}$$

Perioidičnost i sezonalnost

Za sezonske modele s k sezona:

$$\mathbf{n}_{t+k} = \mathbf{A}_k \mathbf{A}_{k-1} \cdots \mathbf{A}_1 \mathbf{n}_t = \mathbf{B}\mathbf{n}_t \tag{3.50}$$

Godišnja stopa rasta: λ_{annual} = dominantna svojstvena vrijednost od $\mathbf{B}$

3.3 Metapopulacijski modeli

3.3.1 Uvod u metapopulacijsku ekologiju

Metapopulacija je skup lokalnih populacija povezanih migracijom. Koncept je uveo Richard Levins 1969. za opisivanje dinamike populacija u fragmentiranim staništima. U metapopulacijskim modelima lokalne populacije mogu izumrijeti, ali ponovno se mogu uspostaviti kolonizacijom iz drugih lokalnih populacija.

Ključni koncepti

Lokalne populacije: Grupe jedinki u diskretnim fragmentima staništa **Migracija:** Kretanje jedinki između lokalnih populacija **Kolonizacija:** Uspostavljanje novih lokalnih populacija **Lokalno izumiranje:** Nestanak lokalnih populacija **Povezanost:** Mogućnost migracije između fragmenata

3.3.2 Levinsov klasični model

Osnovni okvir

Levinsov model tretira dinamiku udjela okupiranih staništa umjesto praćenja brojnosti jedinki. Neka je $p(t)$ udio okupiranih fragmenata u vremenu t.

Osnovna jednadžba:

$$\frac{dp}{dt} = cp(1-p) - ep \tag{3.51}$$

gdje su:

- c - stopa kolonizacije per okupiranog fragmenta
- e - stopa lokalnog izumiranja per okupiranog fragmenta
- p - udio okupiranih fragmenata
- $(1-p)$ - udio neokupiranih fragmenata dostupnih za kolonizaciju

Analiza Levinsovog modela

Ravnotežne točke:
Postavljanjem $dp/dt = 0$:

$$p^* = 0 \quad \text{(uvijek postoji)} \tag{3.52}$$

$$p^* = 1 - \frac{e}{c} \quad \text{(ako } c > e\text{)} \tag{3.53}$$

Uvjeti za opstanak metapopulacije:

$$c > e \quad \text{ili} \quad \frac{c}{e} > 1 \tag{3.54}$$

Omjer c/e naziva se **osnovni reprodukcijski broj metapopulacije** R_0.
Stabilnost ravnoteže:
Linearizacija oko $p^* = 1 - e/c$:

$$\frac{d}{dt}(p - p^*) = -e(p - p^*) \tag{3.55}$$

Ravnoteža je stabilna jer je $e > 0$.

Dinamika Levinsovog modela

Rješenje diferencijalne jednadžbe:
Za $c > e$:

$$p(t) = \frac{(c - e)p_0}{c - e + ep_0 + (c - e - cp_0)e^{-(c-e)t}} \tag{3.56}$$

gdje je $p_0 = p(0)$ početni udio okupiranih fragmenata.
Asimptotsko ponašanje:

$$\lim_{t\to\infty} p(t) = 1 - \frac{e}{c} \quad \text{(ako } c > e\text{)} \tag{3.57}$$

$$\lim_{t\to\infty} p(t) = 0 \quad \text{(ako } c \leq e\text{)} \tag{3.58}$$

3.3.3 Prostorno eksplicitni metapopulacijski modeli

Model s prostornim rasporedom

Prostorna udaljenost utječe na vjerojatnost kolonizacije. Za n fragmenata:

$$\frac{dS_i}{dt} = C_i(1 - S_i) - E_iS_i \tag{3.59}$$

gdje je S_i vjerojatnost da je fragment i okupiran.
Stopa kolonizacije fragmenta i:

$$C_i = \sum_{j\neq i} c_{ij}S_j \tag{3.60}$$

gdje je c_{ij} stopa kolonizacije fragmenta i iz fragmenta j.
Funkcija dispersije:
Najčešće se koristi eksponencijalni kernel:

$$c_{ij} = c_0e^{-d_{ij}/\alpha} \tag{3.61}$$

gdje su:

- d_{ij} - udaljenost između fragmenata i i j
- α - srednja udaljenost dispersije
- c_0 - bazalna stopa kolonizacije

Metapopulacijska kapaciteta

Hanski je uveo koncept metapopulacijske kapacitete λ_M:

$$\lambda_M = \text{dominantna svojstvena vrijednost od } \mathbf{M} \tag{3.62}$$

gdje je $\mathbf{M}$ matrica migracije s elementima:

$$M_{ij} = \begin{cases} \frac{c_{ij}}{E_i} & \text{ako } i \neq j \\ 0 & \text{ako } i = j \end{cases} \tag{3.63}$$

Uvjet za opstanak:

$$\lambda_M > 1 \tag{3.64}$$

3.3.4 Modeli s heterogenošću fragmenata

Varijacije u kvaliteti staništa

Fragmenti mogu imati različite kapacitete, što utječe na stope izumiranja:

$$E_i = \frac{e_0}{A_i^z} \tag{3.65}$$

gdje su:

- A_i - površina fragmenta i
- e_0 - bazalna stopa izumiranja
- z - eksponent površine (obično $z \approx 0.5$)

Teorija biogeografije otoka:
MacArthur-Wilson model povezuje veličinu i izoliranost:

$$C_i = c_0 D_i^{-a} \tag{3.66}$$
$$E_i = e_0 A_i^{-z} \tag{3.67}$$

gdje je D_i mjera izoliranosti fragmenta i.

Rescue efekt

Imigracija može spasiti lokalne populacije od izumiranja:

$$E_i(I_i) = \frac{E_i^0}{1 + \beta I_i} \tag{3.68}$$

gdje je I_i stopa imigracije u fragment i.

3.3.5 Diskretni metapopulacijski modeli

Diskretna verzija Levinsovog modela

$$p_{t+1} = p_t + cp_t(1 - p_t) - ep_t \tag{3.69}$$

ili:

$$p_{t+1} = p_t(1 + c(1 - p_t) - e) \tag{3.70}$$

Mapiranje:

$$p_{t+1} = p_t(1 + c - e - cp_t) = p_t(R_0 - cp_t) \tag{3.71}$$

gdje je $R_0 = (c - e + 1)$.

Stohastički diskretni model

Uključuje demografsku stohastičnost u malim lokalnim populacijama:

$$P(S_{i,t+1} = 1) = S_{i,t}(1 - E_i) + (1 - S_{i,t})C_i \tag{3.72}$$

gdje su $S_{i,t} \in \{0, 1\}$ indikatori okupiranosti.

3.3.6 Fragmentacija staništa

Utjecaj fragmentacije

Fragmentacija utječe na metapopulacijsku dinamiku kroz:

Smanjenje površine staništa:

$$H = \sum_{i=1}^{n} A_i \tag{3.73}$$

Povećanje izoliranosti:

$$I = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} d_{ij} \tag{3.74}$$

Smanjenje povezanosti:

$$\Gamma = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i} e^{-d_{ij}/\alpha} \tag{3.75}$$

Kritični prag fragmentacije

Postoji kritična razina fragmentacije ispod koje metapopulacija ne može postojati:

$$H_{\text{crit}} = \frac{e}{c} \cdot \frac{A_{\text{total}}}{\Gamma_{\max}} \tag{3.76}$$

Perkolacijski model:

Za regularne mreže, kritični prag perkolacije je:

- 2D kvadratna mreža: $p_c \approx 0.593$
- 2D heksagonalna mreža: $p_c = 0.5$
- 2D trokutasta mreža: $p_c = 0.5$

3.3.7 Empirijski primjer: Metapopulacija leptira

Melitaea cinxia na Ålandskim otocima

Hanski i suradnici proučavali su metapopulaciju leptira *Melitaea cinxia*:

Empirijski model:

$$C_i = \frac{c \sum_{j \neq i} S_j e^{-d_{ij}/\alpha}}{1 + c \sum_{j \neq i} S_j e^{-d_{ij}/\alpha}} \tag{3.77}$$

$$E_i = \frac{e_0}{A_i^z} \tag{3.78}$$

Procijenjeni parametri:

- $\alpha \approx 2$ km (srednja udaljenost dispersije)
- $z \approx 0.5$ (eksponent površine-izumiranje)

- $c \approx 1.5$ (kolonizacijski parametar)
- $e_0 \approx 0.3$ (bazalna stopa izumiranja)

Prediktivni uspjeh

Model je uspješno predvidio:

- 70% točnost za kolonizaciju
- 85% točnost za izumiranje
- Dugoročne trendove populacije

3.3.8 Praktične primjene metapopulacijskih modela

Dizajn rezervata

SLOSS debata: Single Large or Several Small

Metapopulacijski modeli pokazuju da optimalni dizajn ovisi o:

- Dispersijskim mogućnostima vrste
- Koreleiranosti lokalnih izumiranja
- Trade-off između veličine i broja fragmenata

Optimizacijski problem:

$$\max_{\{A_i\}} \lambda_M \quad \text{s.t.} \quad \sum A_i \leq A_{\text{total}} \tag{3.79}$$

Minimalna životna populacija

Za metapopulacije, MVP se definira kao:

$$\text{MVP} = \min\{N : P(\text{izumiranje u } T \text{ godina}) < \alpha\} \tag{3.80}$$

gdje se uključuje i prostorna komponenta rizika.

Prostorno eksplicitni MVP:

$$N_{\text{MVP}} = k \cdot \frac{H_{\min}}{\bar{A}} \cdot \bar{n} \tag{3.81}$$

gdje su:

- $H_{\min}$ - minimalna površina staništa
- $\bar{A}$ - srednja površina fragmenta
- $\bar{n}$ - srednja veličina lokalne populacije
- k - sigurnosni faktor (obično $k = 2 - 5$)

Poglavlje 4

Međuspecijske interakcije i modeli zajednica

4.1 Grabljivac–plijen i funkcijski odgovori

4.1.1 Uvod u predator-prey dinamiku

Interakcije između grabljivaca i plijena temeljne su za razumijevanje dinamike ekoloških zajednica. Ove interakcije karakteriziraju se složenim povratnim spregama: povećanje broja plijena omogućava rast populacije grabljivaca, što dovodi do smanjenja populacije plijena, što zauzvrat smanjuje populaciju grabljivaca.

Klasifikacija trofnih interakcija

Tablica 4.1: Klasifikacija međuspecijskih interakcija

Interakcija	**Sp. 1**	**Sp. 2**	**Primjer**
Predacija	+	−	Lav-zebra
Parazitizam	+	−	Virus-domaćin
Mutualism	+	+	Pčela-cvijet
Komenzalizam	+	0	Morski pas-riba
Amenzalizam	−	0	Alelopatija
Konkurencija	−	−	Dva predatora
Neutralizam	0	0	Udaljene vrste

4.1.2 Lotka-Volterra modeli

Osnovni Lotka-Volterra model

Alfredo Lotka (1925) i Vito Volterra (1926) neovisno su razvili prvi matematički model predator-prey dinamike. Model pretpostavlja:

- Eksponencijalni rast plijena u odsutnosti predatora
- Stopa predacije proporcionalna umnošku abundancija
- Smrt predatora exponencijalna u odsutnosti plijena
- Konverzijska efikasnost konstanta

Osnovni sustav jednadžbi:

$$\frac{dN}{dt} = rN - aNP \tag{4.1}$$
$$\frac{dP}{dt} = eaNP - mP \tag{4.2}$$

gdje su:

- $N(t)$ - gustoća populacije plijena
- $P(t)$ - gustoća populacije predatora
- r - intrinsična stopa rasta plijena
- a - stopa napada predatora
- e - efikasnost konverzije ($0 < \mathrm{e} < 1$)
- m - stopa smrti predatora

Analiza ravnotežnih točaka

Nullclines:

Postavljanjem derivacija na nulu:

$$\frac{dN}{dt} = 0: \quad N(r - aP) = 0 \tag{4.3}$$
$$\frac{dP}{dt} = 0: \quad P(eaN - m) = 0 \tag{4.4}$$

Ravnotežne točke:

$$\text{Trivijalna: } (N^*, P^*) = (0, 0) \tag{4.5}$$
$$\text{Koegzistencijska: } (N^*, P^*) = \left(\frac{m}{ea}, \frac{r}{a}\right) \tag{4.6}$$

Linearna stabilnostna analiza

Jacobian matrica oko koegzistencijske ravnoteže:

$$\mathbf{J} = \begin{pmatrix} 0 & -aN^* \\ eaP^* & 0 \end{pmatrix} = \begin{pmatrix} 0 & -m/e \\ er & 0 \end{pmatrix} \tag{4.7}$$

Karakteristična jednadžba:

$$\det(\mathbf{J} - \lambda\mathbf{I}) = \lambda^2 + \frac{mer}{e} = \lambda^2 + mr = 0 \tag{4.8}$$

Svojstvene vrijednosti:

$$\lambda_{1,2} = \pm i\sqrt{mr} \tag{4.9}$$

Čisto imaginarni svojstvene vrijednosti označavaju **neutralno stabilnu ravnotežu** s periodičnim oscilacijama.

Konzervirana količina

Lotka-Volterra sustav ima konzerviraną količinu (Hamiltonijan):

$$H(N,P) = eaN - m\ln(N) + aP - r\ln(P) = \text{konstanta} \tag{4.10}$$

Izvod:

$$\frac{dH}{dt} = \frac{\partial H}{\partial N}\frac{dN}{dt} + \frac{\partial H}{\partial P}\frac{dP}{dt} \tag{4.11}$$

$$= \left(ea - \frac{m}{N}\right)(rN - aNP) + \left(a - \frac{r}{P}\right)(eaNP - mP) \tag{4.12}$$

$$= 0 \tag{4.13}$$

Perioda oscilacija

Period oscilacija oko ravnoteže:

$$T = \frac{2\pi}{\sqrt{mr}} \tag{4.14}$$

Amplitude oscilacija ovise o početnim uvjetima i određene su konzerviranom količinom.

4.1.3 Funkcijski odgovori

Definicija i tipovi

Funkcijski odgovor opisuje vezu između gustoće plijena i stope konzumacije po predatoru. C.S. Holling (1959) identificirao je tri osnovna tipa:

Tip I - Linearan:

$$f(N) = aN \tag{4.15}$$

Tip II - Hiperbolički (Michaelis-Menten):

$$f(N) = \frac{aN}{1 + ahN} \tag{4.16}$$

Tip III - Sigmoidalni:

$$f(N) = \frac{aN^2}{1 + ahN^2} \tag{4.17}$$

gdje je h handling time (vrijeme potrebno za hvatanje i konzumaciju jedne jedinke plijena).

Izvod Tip II funkcijskog odgovora

Pretpostavke:

- Ukupno vrijeme = traženja + handling
- $T = T_s + T_h$
- Stopa susreta proporcionalna N: attack rate = a
- Handling time po plijenu: h

Izvod:

Broj uhvaćenih jedinki plijena u vremenu T:

$$N_e = aT_sN \tag{4.18}$$

Vrijeme handling:

$$T_h = hN_e \tag{4.19}$$

Ukupno vrijeme:

$$T = T_s + hN_e = T_s + haT_sN \tag{4.20}$$

Rješavanjem za T_s:

$$T_s = \frac{T}{1 + haN} \tag{4.21}$$

Stopa konzumacije:

$$f(N) = \frac{N_e}{T} = \frac{aT_sN}{T} = \frac{aN}{1 + ahN} \tag{4.22}$$

Svojstva funkcijskih odgovora

Tip I:

- Linearna veza do zasićenja
- $f'(N) = a > 0$ (konstanta)
- Nema handling time ograničenja
- Rijetko u prirodi

Tip II:

- $\lim_{N\to\infty} f(N) = 1/h$ (maksimalna stopa)
- $f'(N) = \frac{a}{(1+ahN)^2} > 0$ (uvijek pozitivna derivacija)
- $f''(N) = \frac{-2a^2hN}{(1+ahN)^3} < 0$ (konkavna)
- Može destabilizirati predator-prey dinamiku

Tip III:

- S-oblik s infleksijskim točkom
- $f'(0) = 0$ (nulta derivacija kod $N = 0$)
- Infleksijska točka kod $N^* = 1/(\sqrt{ah})$
- Omogućava refuge za plijen pri niskim gustoćama
- Stabilizira predator-prey dinamiku

4.1.4 Modifikacije Lotka-Volterra modela

Model s Tip II funkcijskim odgovorom

$$\frac{dN}{dt} = rN - \frac{aNP}{1 + ahN} \tag{4.23}$$

$$\frac{dP}{dt} = \frac{eaNP}{1 + ahN} - mP \tag{4.24}$$

Ravnotežne točke:
Koegzistencijska ravnoteža:

$$N^* = \frac{m}{ea - mah} \tag{4.25}$$

$$P^* = \frac{r(1 + ahN^*)}{a} \tag{4.26}$$

Uvjet za postojanje:

$$ea > mah \quad \text{ili} \quad \frac{e}{h} > m \tag{4.27}$$

Rosenzweig-MacArthur model

Dodavanje logističkog rasta plijena:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) - \frac{aNP}{1 + ahN} \tag{4.28}$$

$$\frac{dP}{dt} = \frac{eaNP}{1 + ahN} - mP \tag{4.29}$$

Paradoks obogaćivanja:
Povećanje K (nosivost staništa) može destabilizirati sustav i dovesti do ekstremnih oscilacija ili izumiranja predatora.

Model s Tip III funkcijskim odgovorom

$$\frac{dN}{dt} = rN - \frac{aN^2P}{1 + ahN^2} \tag{4.30}$$

$$\frac{dP}{dt} = \frac{eaN^2P}{1 + ahN^2} - mP \tag{4.31}$$

Stabiliziranje svojstvo:
Tip III funkcijski odgovor stvara **refuge efekt** za plijen pri niskim gustoćama, što stabilizira dinamiku.

4.1.5 Prostorni predator-prey modeli

Reakcijska-difuzija modeli

$$\frac{\partial N}{\partial t} = rN - aNP + D_N\nabla^2 N \tag{4.32}$$

$$\frac{\partial P}{\partial t} = eaNP - mP + D_P\nabla^2 P \tag{4.33}$$

gdje su D_N i D_P difuzijski koeficijenti.

Prostorna stabilizacija

Prostorna heterogenost može stabilizirati inače nestabilne predator-prey dinamike kroz:

- **Asinkrone oscilacije** između lokaliteta
- **Source-sink dinamiku**
- **Prostorne refugije** za plijen

4.2 Konkurencija i koegzistencija

4.2.1 Teorija konkurencije

Tipovi konkurencije

Interference konkurencija: Direktne agresivne interakcije
Exploitation konkurencija: Konkurencija za ograničene resurse
Apparent konkurencija: Posredovana zajedničkim predatorima

Gause-ov princip

G.F. Gause (1934) postulirao je **competitive exclusion principle**:

"Dvije vrste s identičnim ekološkim potrebama ne mogu dugoročno koegzistirati u istom staništu."

4.2.2 Lotka-Volterra model konkurencije

Dvospecijski model

$$\frac{dN_1}{dt} = r_1 N_1 \left(1 - \frac{N_1 + \alpha_{12} N_2}{K_1}\right) \tag{4.34}$$

$$\frac{dN_2}{dt} = r_2 N_2 \left(1 - \frac{N_2 + \alpha_{21} N_1}{K_2}\right) \tag{4.35}$$

gdje su:

- α_{12} - utjecaj vrste 2 na vrstu 1 (competition coefficient)
- α_{21} - utjecaj vrste 1 na vrstu 2
- K_1, K_2 - nosivosti staništa kada su vrste same

Isokline i fazni portret

Nullclines (isokline):

$$\frac{dN_1}{dt} = 0: \quad N_1 = K_1 - \alpha_{12} N_2 \tag{4.36}$$

$$\frac{dN_2}{dt} = 0: \quad N_2 = K_2 - \alpha_{21} N_1 \tag{4.37}$$

Ravnotežne točke:

1. $(0, 0)$ - nestabilna
2. $(K_1, 0)$ - vrste 1 pobjeđuje
3. $(0, K_2)$ - vrste 2 pobjeđuje
4. (N_1^*, N_2^*) - koegzistencija (ako postoji)

Koegzistencijska ravnoteža:

$$N_1^* = \frac{K_1 - \alpha_{12} K_2}{1 - \alpha_{12}\alpha_{21}} \tag{4.38}$$

$$N_2^* = \frac{K_2 - \alpha_{21} K_1}{1 - \alpha_{12}\alpha_{21}} \tag{4.39}$$

Uvjeti za koegzistenciju

Uvjet postojanja:

$$N_1^*, N_2^* > 0 \tag{4.40}$$

To zahtijeva:

$$K_1 > \alpha_{12} K_2 \tag{4.41}$$

$$K_2 > \alpha_{21} K_1 \tag{4.42}$$

Uvjet stabilnosti:
Jacobian oko koegzistencijske ravnoteže:

$$\mathbf{J} = \begin{pmatrix} -\frac{r_1 N_1^*}{K_1} & -\frac{r_1 \alpha_{12} N_1^*}{K_1} \\ -\frac{r_2 \alpha_{21} N_2^*}{K_2} & -\frac{r_2 N_2^*}{K_2} \end{pmatrix} \tag{4.43}$$

Uvjeti stabilnosti (Routh-Hurwitz):

$$\text{tr}(\mathbf{J}) < 0 \quad \text{(uvijek zadovoljeno)} \tag{4.44}$$

$$\det(\mathbf{J}) > 0 \tag{4.45}$$

Uvjet $\det(\mathbf{J}) > 0$ ekvivalentan je:

$$\alpha_{12}\alpha_{21} < 1 \tag{4.46}$$

Scenariji konkurencije

1. Stabilna koegzistencija:

$$\alpha_{12} < \frac{K_1}{K_2} \quad \text{i} \quad \alpha_{21} < \frac{K_2}{K_1} \tag{4.47}$$

2. Vrste 1 uvijek pobjeđuje:

$$\alpha_{12} > \frac{K_1}{K_2} \quad \text{i} \quad \alpha_{21} < \frac{K_2}{K_1} \tag{4.48}$$

3. Vrste 2 uvijek pobjeđuje:

$$\alpha_{12} < \frac{K_1}{K_2} \quad \text{i} \quad \alpha_{21} > \frac{K_2}{K_1} \tag{4.49}$$

4. Prioritetni efekt (bistabilnost):

$$\alpha_{12} > \frac{K_1}{K_2} \quad \text{i} \quad \alpha_{21} > \frac{K_2}{K_1} \tag{4.50}$$

4.2.3 Moderna teorija koegzistencije

Chesson okvir

Peter Chesson razvio je sveobuhvatan okvir koji dijeli mehanizme koegzistencije na:
Stabilizarne mehanizme: Povećavaju negativnu povratnu spregu **Equaliziranje mehanizmi:** Smanjuju fitness razlike između vrsta

Invazijski rast

Vrsta i može invadirati zajednicu ako je njena dugoročna stopa rasta pozitivna kada je rijetka:

$$\bar{r}_i = r_i - \sum_{j \neq i} \alpha_{ij} \bar{N}_j > 0 \tag{4.51}$$

Recipročna invazivnost - uvjet za stabilnu koegzistenciju.

Modern coexistence mechanisms

1. Storage efekt:

$$\text{Kovarijanca}(\text{rast}, \text{konkurencija}) < 0 \tag{4.52}$$

Koegzistencija zbog vremenskih varijacija u uvjetima.

2. Speed-accuracy trade-off:

Kompromis između brzine rasta i efikasnosti korištenja resursa.

3. Relative nonlinearity:

Nelinearne interakcije koje pogoduju vrsti kada je rijetka.

4.2.4 Resource partitioning i niche teorija

MacArthur-ov model resursa

$$\frac{dN_i}{dt} = N_i \left(\sum_{j=1}^{m} c_{ij} R_j - d_i \right) \tag{4.53}$$

$$\frac{dR_j}{dt} = S_j - \sum_{i=1}^{n} c_{ij} N_i R_j \tag{4.54}$$

gdje su:

- R_j - abundancija resursa j
- c_{ij} - consumption rate vrste i na resurs j
- S_j - supply rate resursa j
- d_i - death rate vrste i

Limiting similarity

MacArthur i Levins (1967) pokazali su da postoji minimalna razlika između niša potrebna za koegzistenciju.

Gaussian utiliziranje resursa:

$$u_i(x) = \exp\left(-\frac{(x - \mu_i)^2}{2\sigma^2} \right) \tag{4.55}$$

Uvjet koegzistencije:

$$\frac{|\mu_1 - \mu_2|}{\sigma} > \sigma_{\min} \tag{4.56}$$

gdje je $\sigma_{\min} \approx 1$ za Gaussian niše.

Character displacement

Evolucijski odgovor na konkurenciju koji povećava razlike u trait-ima.

Quantitative genetski model:

$$\frac{d\bar{z}_i}{dt} = h_i^2 \frac{\partial \bar{w}_i}{\partial \bar{z}_i} \tag{4.57}$$

gdje je h_i^2 heritabilnost, $\bar{z}_i$ srednji trait, i $\bar{w}_i$ fitness.

4.3 Ekološke mreže i stabilnost zajednica

4.3.1 Uvod u ekološke mreže

Ekološke zajednice najbolje se razumiju kao složene mreže interakcija između vrsta. Mrežna analiza omogućava kvantificiranje strukture zajednica i ispitivanje veze između kompleksnosti i stabilnosti.

Tipovi ekoloških mreža

Food webs: Tko koga jede (trofne interakcije) **Mutualistička mreže:** Beneficial interakcije (oprašivanje, rasipanje sjemenki) **Host-parazitska mreže:** Parazitsko-host veze **Mreže za čišćenje:** Cleaning symbioses

4.3.2 Topologija hranidbenih mreža

Osnovni parametri mreže

Connectance (C):

$$C = \frac{L}{S^2} \tag{4.58}$$

gdje je L broj veza, S broj vrsta.
Link density:

$$\frac{L}{S} \tag{4.59}$$

Degree distribucija:

$$P(k) = \text{vjerojatnost da vrsta ima } k \text{ veza} \tag{4.60}$$

Trofni nivoi

Shortweighted trophic level:

$$\mathrm{TL}_i = 1 + \frac{\sum_j \mathrm{TL}_j \cdot w_{ji}}{\sum_j w_{ji}} \tag{4.61}$$

gdje je w_{ji} težina veze od vrste j do vrste i.
Omnivory index:

$$\mathrm{OI}_i = \sum_j (\mathrm{TL}_j - \mathrm{TL}_i + 1)^2 \cdot \frac{w_{ji}}{\sum_k w_{ki}} \tag{4.62}$$

Modularnost

Newman-ova modularnost:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{4.63}$$

gdje su:

- A_{ij} - adjacency matrica
- k_i - degree čvor i
- m - ukupni broj veza
- c_i - zajednica čvor i
- $\delta(c_i, c_j) = 1$ ako $c_i = c_j$, inače 0

4.3.3 May-ova analiza stabilnosti

Random matrix pristup

Robert May (1972) koristio je random matrix teoriju za analizu stabilnosti velikih ekoloških mreža.

Model:

$$\frac{dx_i}{dt} = x_i \left(r_i + \sum_{j=1}^{S} a_{ij} x_j \right) \tag{4.64}$$

Linearizacija oko ravnoteže:

$$\frac{d\xi_i}{dt} = \sum_{j=1}^{S} A_{ij} \xi_j \tag{4.65}$$

gdje je $A_{ij} = a_{ij} x_j^*$.

Random matrica svojstva

Elementi community matrice:

- $A_{ii} = -1$ (self-regulation)
- $A_{ij} \sim N(0, \sigma^2)$ with probability C
- $A_{ij} = 0$ with probability $(1 - C)$

May-ov stabilnostni uvjet:

Za velike mreže, uvjet stabilnosti je:

$$\sigma\sqrt{SC} < 1 \tag{4.66}$$

ili ekvivalentno:

$$\sigma\sqrt{L} < 1 \tag{4.67}$$

Interpretacija May-ovog rezultata

May-ov rezultat sugerira da kompleksnost (veći S ili C) destabilizira zajednice, što je kontradiktorno empirijskim opažanjima.

Paradoks složenosti-stabilnosti:

- Prirodne zajednice su složene i stabilne - Random modeli predviđaju da složenost destabilizira - Razlika nastaje zbog struktura koja nije random

4.3.4 Strukturni aspekti stabilnosti

Trofna koherentnost

Kramer i suradnici pokazali su da trofno koherentne mreže (s jasnom hijerarhijskom strukturom) mogu biti stabilne unatoč velikoj kompleksnosti.

Trofna koherentnost:

$$q = \frac{1}{L} \sum_{i \to j} |h_i - h_j - 1|^2 \tag{4.68}$$

gdje je h_i trofni nivo vrste i.

Modularna struktura

Zajednice organizirane u module stabilnije su od random mreža.

Within-module vs. between-module veze:

$$\frac{\sigma_{\text{within}}}{\sigma_{\text{between}}} \tag{4.69}$$

4.3.5 Kaskadni učinci

Trofni kaskadni učinci

Promjene na vrhu hranidbene mreže mogu se propagirati prema dolje kroz trofne nivoe.

Klasični trofni kaskadni učinak:

$$\text{Top predatori} \uparrow \Rightarrow \text{Mesopredatori} \downarrow \Rightarrow \text{Herbivori} \uparrow \Rightarrow \text{Primarni proizvođači} \downarrow \tag{4.70}$$

Sekundarni izumiranje

Izumiranje jedne vrste može dovesti do kaskadnih izumiranja.

Robusnost mreže:

$$R = \frac{S - S_{\text{extinct}}}{S} \tag{4.71}$$

gdje je S_{extinct} broj vrsta koje izumiru nakon perturbacije.

Keystone vrste

Vrste čije uklanjanje ima disproportionalno veliki utjecaj na strukturu zajednice.

Keystoneness index:

$$K_i = \frac{\Delta D_i}{\Delta B_i} \tag{4.72}$$

gdje je ΔD_i promjena u raznolikosti, ΔB_i promjena u biomasi vrste i.

4.3.6 Dinamička analiza mreža

Perturbation analiza

Press perturbation:

Trajna promjena u parametru:

$$\Delta \mathbf{x}^* = -\mathbf{A}^{-1} \Delta \mathbf{r} \tag{4.73}$$

Pulse perturbation:

Trenutačna promjena u gustoći:

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{x}(0) \tag{4.74}$$

Return time

Vrijeme potrebno za povratak na ravnotežu nakon perturbacije:

$$T_{\text{return}} = -\frac{1}{\text{Re}(\lambda_{\max})} \tag{4.75}$$

gdje je $\lambda_{\max}$ najveća (najmanje negativna) svojstvena vrijednost.

4.3.7 Empirijski primjeri

Yellowstone vukovi

Reintrodukcija vukova u Yellowstone (1995) dovela je do:

- Smanjenja populacije jelena
- Oporavka vegetacije (vrbe, topole)
- Povratka bobera
- Promjena tijeka rijeka (zbog vegetacije)

Trofomorfic cascades: Predatori oblikuju fizičku strukturu ekosustava.

Morska vidra i kelp šume

Interakcijski lanac:

$$\text{Morska vidra} \rightarrow \text{Morski ježevi} \rightarrow \text{Kelp alge} \tag{4.76}$$

Gubitak morskih vidara doveo je do: - Eksplozije populacije morskih ježeva - Degradacije kelp šuma - Gubitka staništa za mnoge vrste

4.3.8 Mrežne metrike i bioraznolikost

Species importance indices

Betweenness centrality:

$$g_i = \sum_{s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{4.77}$$

gdje je $\sigma_{st}(i)$ broj najkraćih putanja između s i t koji prolaze kroz i.

Closeness centrality:

$$c_i = \frac{1}{\sum_{j \neq i} d_{ij}} \tag{4.78}$$

gdje je d_{ij} najkraća udaljenost između i i j.

Network motifs

Mali subgrafovi koji se javljaju često:

3-node motifs:

- Omnivory loop
- Apparent competition
- Three-species chain

Z-score:

$$Z = \frac{N_{\text{real}} - N_{\text{random}}}{\sigma_{\text{random}}} \tag{4.79}$$

4.3.9 Prostorno-eksplicitni mrežni modeli

Metahranidbene mreže

Kombinacija metapopulacijske dinamike i trofnih interakcija:

$$\frac{dS_{ij}}{dt} = C_{ij}(1 - S_{ij}) - E_{ij}S_{ij} \tag{4.80}$$

gdje S_{ij} označava okupiranost vrste i u patch-u j.

Prostorna stabilnost hranidbenih mreža

Prostorna struktura može stabilizirati hranidbene mreže kroz:

- **asinkronu dinamiku** između zakrpa staništa
- **rekolonizaciju** nakon lokalnih izumiranja
- **prostorne dotoke** između staništa

Prostorni stabilizacijski uvjet:

$$\sigma\sqrt{SC} < \sqrt{1 + \frac{D}{\sigma^2}} \tag{4.81}$$

gdje je D stopa disperzije.

Poglavlje 5

Modeli ekosustava i bioenergetika

5.1 Kruženje tvari i energije

5.1.1 Osnovni principi i bilance tvari

Ekosustavi funkcioniraju kao složeni sustavi koji obrađuju energiju i materijale prema fundamentalnim fizikalnim zakonima. Prvi zakon termodinamike nalaže da se energija ne može stvarati niti uništavati, već se samo može transformirati iz jednog oblika u drugi. Za bilo koji ekosustav možemo postaviti temeljnu jednadžbu bilance:

$$\frac{dS}{dt} = I - O - T - D \tag{5.1}$$

gdje je:

- S = zaliha tvari ili energije u sustavu
- I = ulazni tok (input)
- O = izlazni tok (output)
- T = transformacija unutar sustava
- D = degradacija ili disipacija

5.1.2 Biogeokemijski ciklusi

Ciklus ugljika

Model ugljičnog ciklusa može se opisati sustavom diferencijalnih jednadžbi za različite rezervoare. Za jednostavan model s tri rezervoara (atmosfera, biomasa, tlo):

$$\frac{dC_a}{dt} = \gamma \cdot R_{soil} + \gamma \cdot R_{bio} - GPP + F_{fossil} \tag{5.2}$$

$$\frac{dC_{bio}}{dt} = (1 - \alpha) \cdot GPP - R_{bio} - M \tag{5.3}$$

$$\frac{dC_{soil}}{dt} = \alpha \cdot GPP + M - R_{soil} \tag{5.4}$$

gdje je:

- C_a, C_{bio}, C_{soil} = ugljik u atmosferi, biomasi i tlu
- GPP = bruto primarna produktivnost (Gross Primary Productivity)
- R_{bio}, R_{soil} = respiracija biomase i tla

- α = frakcija GPP koja ide u tlo
- M = smrtnost biomase
- γ = konverzijski faktor CO C
- F_{fossil} = emisije iz fosilnih goriva

Respiracija se često modelira kao funkcija temperature koristeći Q odnos:

$$R(T) = R_{ref} \cdot Q_{10}^{(T-T_{ref})/10} \tag{5.5}$$

gdje je tipično $Q_{10} = 2 - 3$.

Ciklus dušika

Dušični ciklus uključuje multiple kemijske oblike. Osnovni model uključuje:

$$\frac{dNH_4^+}{dt} = k_{min} \cdot N_{org} - k_{nit} \cdot NH_4^+ - U_{NH4} \tag{5.6}$$

$$\frac{dNO_3^-}{dt} = k_{nit} \cdot NH_4^+ - k_{denit} \cdot NO_3^- - U_{NO3} \tag{5.7}$$

$$\frac{dN_{org}}{dt} = -k_{min} \cdot N_{org} + I_{org} + D_{mort} \tag{5.8}$$

gdje su:

- $k_{min}, k_{nit}, k_{denit}$ = konstante brzine mineralizacije, nitrifikacije i denitrifikacije
- U_{NH4}, U_{NO3} = biljno usvajanje amonija i nitrata
- I_{org} = ulaz organske tvari
- D_{mort} = dušik iz smrtnosti biomase

5.1.3 Energetski tokovi kroz ekosustav

Energetski tok kroz trofične razine slijedi eksponencijalno slabljenje:

$$E_n = E_0 \cdot \epsilon^n \tag{5.9}$$

gdje je:

- E_n = energija na trofičnoj razini n
- E_0 = energija primarnih producenata
- ϵ = efikasnost transfera (tipično 0.1-0.2)

Lindeman-ova piramida efikasnosti

Efikasnost transfera energije između trofičnih razina:

$$TE_n = \frac{P_n}{P_{n-1}} = \frac{A_n \cdot AE_n \cdot PE_n}{P_{n-1}} \tag{5.10}$$

gdje je:

- TE_n = efikasnost transfera na razinu n
- P_n = produkcija na razini n

- A_n = asimilacija na razini n
- AE_n = efikasnost asimilacije
- PE_n = efikasnost produkcije

5.1.4 Ohmov zakon za ekologiju

Po analogiji s Ohmovim zakonom u elektrotehnici, možemo definirati "otpor" ekosustava prema toku tvari:

$$J = \frac{\Delta C}{R_{eco}} \tag{5.11}$$

gdje je:

- J = tok tvari
- ΔC = razlika koncentracija
- R_{eco} = ekološki otpor

5.2 Bioenergetski i DEB modeli

5.2.1 Osnove DEB teorije

Dynamic Energy Budget (DEB) teorija pruža kvantitativni okvir za metabolizam organizma tijekom cijelog životnog ciklusa. Temelji se na nekoliko ključnih pretpostavki:

1. **Kappa pravilo**: Energija se dijeli između rasta/održavanja (κ) i reprodukcije/sazrijevanja ($(1-\kappa)$)
2. **Surface area law**: Brzina hranjenja proporcionalna je površini
3. **Volume-specific somatic maintenance**: Održavanje je proporcionalno volumenu

Osnovne varijable stanja

DEB model koristi tri varijable stanja:

- E = energija u rezervi (J)
- V = strukturalni volumen (cm^3)
- E_H = energija uložena u sazrijevanje (J)

Temeljne jednadžbe

Brzina hranjenja:

$$\dot{p}_X = \{p_{Xm}\} \cdot f \cdot V^{2/3} \tag{5.12}$$

gdje je:

- $\{p_{Xm}\}$ = maksimalna specifična brzina hranjenja
- f = funkcionalni odgovor (0 f 1)
- $V^{2/3}$ = površina

Asimilacija:

$$\dot{p}_A = \{p_{Am}\} \cdot f \cdot V^{2/3} \tag{5.13}$$

Mobilizacija iz rezerve:

$$\dot{p}_C = \frac{E \cdot \dot{v}}{V} \tag{5.14}$$

gdje je $\dot{v}$ = energijska provodljivost.
Kappa pravilo za alokaciju:

$$\dot{p}_G = \kappa \cdot \dot{p}_C - \dot{p}_M \tag{5.15}$$
$$\dot{p}_R = (1-\kappa) \cdot \dot{p}_C - \dot{p}_J \tag{5.16}$$

gdje su:

- $\dot{p}_G$ = rast
- $\dot{p}_R$ = reprodukcija
- $\dot{p}_M$ = somatsko održavanje = $[\dot{p}_M] \cdot V$
- $\dot{p}_J$ = sazrijevanje

Diferencijalne jednadžbe DEB modela

Rezerva energije:

$$\frac{dE}{dt} = \dot{p}_A - \dot{p}_C \tag{5.17}$$

Strukturalni volumen:

$$\frac{dV}{dt} = \frac{\dot{p}_G}{[E_G]} \tag{5.18}$$

gdje je $[E_G]$ = specifična cijena strukture.
Energija sazrijevanja:

$$\frac{dE_H}{dt} = \dot{p}_J = (1-\kappa) \cdot \dot{p}_C - \dot{p}_J \tag{5.19}$$

5.2.2 Standardni DEB model

Za standardni DEB model možemo izvesti analitička rješenja u slučaju konstantnih uvjeta:
Ultimativna duljina:

$$L_\infty = \frac{\{p_{Am}\} \cdot f}{[\dot{p}_M] \cdot \dot{v}} \cdot \delta_M \tag{5.20}$$

von Bertalanffy-jeva jednadžba rasta:

$$L(t) = L_\infty \cdot (1 - e^{-k \cdot t}) \tag{5.21}$$

gdje je von Bertalanffy-jeva konstanta:

$$k = \frac{\dot{v}}{3 \cdot \delta_M \cdot L_\infty} \tag{5.22}$$

5.2.3 Funkcionalni odgovor u DEB kontekstu

Funkcionalni odgovor povezuje gustoću hrane s brzinom hranjenja:

$$f = \frac{X}{X + K} \tag{5.23}$$

gdje je:

- X = gustoća hrane
- K = polu-saturacijska konstanta

Za više tipova hrane:

$$f_i = \frac{X_i}{X_i + K_i} \cdot \prod_{j \neq i} \frac{K_j}{X_j + K_j} \tag{5.24}$$

5.2.4 Povezivanje DEB modela s populacijskim dinamikama

DEB modeli se mogu proširiti na populacijsku razinu kroz strukturirane populacijske modele:

$$\frac{\partial n(a,t)}{\partial t} + \frac{\partial n(a,t)}{\partial a} = -\mu(a, E(a)) \cdot n(a,t) \tag{5.25}$$

s rubnim uvjetom:

$$n(0,t) = \int_0^\infty R(a, E(a)) \cdot n(a,t)\, da \tag{5.26}$$

gdje su:

- $n(a,t)$ = gustoća jedinki dobi a u vremenu t
- $\mu(a, E(a))$ = stopa smrtnosti ovisna o dobi i energetskom stanju
- $R(a, E(a))$ = stopa reprodukcije

5.3 Ecosystem services

5.3.1 Klasifikacija i kvantifikacija

Usluge ekosustava dijele se u četiri glavne kategorije:

1. **Pružateljske usluge** (hrana, voda, drvo)
2. **Regulacijske usluge** (klimatska regulacija, čišćenje vode)
3. **Kulturne usluge** (rekreacija, estetska vrijednost)
4. **Potporne usluge** (kruženje hranjivih tvari, fotosinteza)

Kvantifikacija pomoću produkcijskih funkcija

Opća forma produkcijske funkcije za uslugu ekosustava:

$$ES = f(B_1, B_2, ..., B_n, C_1, C_2, ..., C_m) \tag{5.27}$$

gdje su:

- ES = razina usluge ekosustava
- B_i = biotički čimbenici
- C_j = abiotički čimbenici

Primjer - Sekvestracija ugljika:

$$C_{seq} = \alpha \cdot LAI \cdot PAR \cdot LUE \cdot (1 - R_a - R_h) \tag{5.28}$$

gdje je:

- LAI = indeks listne površine
- PAR = fotosintetski aktivno zračenje
- LUE = efikasnost korištenja svjetlosti
- R_a, R_h = autotrófna i heterotrófna respiracija

5.3.2 Ekonomska vrednovanje

Metoda zamjenskih troškova

Za uslugu čišćenja vode:

$$V_{water} = C_{treatment} \cdot V_{water} \cdot E_{removal} \tag{5.29}$$

gdje je:

- V_{water} = ekonomska vrijednost
- $C_{treatment}$ = trošak tehnološke obrade
- V_{water} = volumen vode
- $E_{removal}$ = efikasnost uklanjanja onečišćenja

Hedonistic pricing

Cijena nekretnine kao funkcija usluga ekosustava:

$$P = \alpha + \sum_i \beta_i X_i + \sum_j \gamma_j ES_j + \epsilon \tag{5.30}$$

gdje su:

- P = cijena nekretnine
- X_i = strukturne karakteristike
- ES_j = usluge ekosustava
- γ_j = marginalne vrijednosti usluga

5.3.3 Trade-off analize

Pareto granica

Za dvije usluge ekosustava možemo definirati Pareto granicu:

$$ES_2^{max} = f(ES_1) \quad \text{subject to} \quad g(ES_1, ES_2) \leq 0 \tag{5.31}$$

Optimizacijski problem:

$$\max \sum_i w_i \cdot ES_i \tag{5.32}$$

$$\text{subject to:} \quad \sum_j a_{ij} x_j \leq b_i \tag{5.33}$$

gdje su:

- w_i = težine usluga
- x_j = upravljačke varijable
- a_{ij}, b_i = ograničenja

Elastičnost supstitucije

Elastičnost supstitucije između dvije usluge:

$$\sigma = \frac{d\ln(ES_2/ES_1)}{d\ln(MRS)} \tag{5.34}$$

gdje je MRS = marginalna stopa supstitucije.

5.3.4 Prostorno modeliranje usluga

InVEST modeli

Model opskrbe vodom:

$$Y(x) = (1 - AET(x)/P(x)) \cdot P(x) \tag{5.35}$$

gdje je:
- $Y(x)$ = godišnja opskrba vodom u pikselu x
- $AET(x)$ = stvarna evapotranspiracija
- $P(x)$ = oborine

Model polinizacije:

$$PS_j = \sum_n \frac{A_n}{1 + (D_{jn}/\alpha_n)^2} \cdot e^{-D_{jn}/\alpha_n} \tag{5.36}$$

gdje je:
- PS_j = usluga polinizacije na lokaciji j
- A_n = površina staništa n
- D_{jn} = udaljenost između j i n
- α_n = prosječna udaljenost leta za vrstu n

5.3.5 Dinamičko modeliranje usluga

Za usluge koje se mijenjaju kroz vrijeme:

$$\frac{dES_i}{dt} = r_i \cdot ES_i \cdot \left(1 - \frac{ES_i}{K_i}\right) - h_i \cdot ES_i - \sum_{j \neq i} \alpha_{ij} \cdot ES_j \tag{5.37}$$

gdje je:
- r_i = intrinsična stopa rasta usluge i
- K_i = nosivost za uslugu i
- h_i = stopa žetve/korištenja
- α_{ij} = interakcijski koeficijenti

Optimalno korištenje kroz vrijeme

Hamiltonijan za optimalno korištenje:

$$H = \sum_i \left[U_i(h_i) + \lambda_i \left(f_i(ES_i) - h_i\right)\right] \tag{5.38}$$

Prvi red uvjeti:

$$\frac{\partial U_i}{\partial h_i} = \lambda_i \tag{5.39}$$

$$\dot{\lambda_i} = \rho\lambda_i - \lambda_i \frac{\partial f_i}{\partial ES_i} \tag{5.40}$$

gdje je ρ = diskontna stopa.

Ovaj sveobuhvatan pristup modeliranju ekosustava i bioenergetike omogućuje kvantitativno razumijevanje složenih interakcija između biotičkih i abiotičkih komponenti, što je ključno za održivo upravljanje prirodnim resursima i uslugama ekosustava.

Dio III

Prostor, predviđanje i AI

Poglavlje 6

Prostorno-ekološko modeliranje

Prostorno-ekološko modeliranje predstavlja jedan od najdinamičnijih i najbrže rastućih područja suvremene ekologije. Ovo interdisciplinarno polje spaja teoretsku ekologiju, geomatiku, statistiku i računalne znanosti kako bi razumjelo i predvidjelo prostorne obrasce distribucije organizama, njihove interakcije s okolišem te procese koji oblikuju bioraznolikost na različitim prostornim skalama.

6.1 GIS, daljinska istraživanja i prostorne skale

6.1.1 Geografski informacijski sustavi (GIS) u ekologiji

Geografski informacijski sustavi predstavljaju tehnološku osnovu prostorno-ekološkog modeliranja. GIS omogućuje pohranu, manipulaciju, analizu i vizualizaciju prostornih podataka, što je nezamjenjivo za razumijevanje ekoloških procesa koji se odvijaju u prostoru i vremenu.

Osnovni principi GIS-a u ekologiji:

1. **Rasterski podatci**: Prostor je podijeljen u mrežu pravokutnih ćelija (piksela), gdje svaka ćelija nosi informaciju o određenoj varijabli (temperatura, padaline, nadmorska visina, tip vegetacije).
2. **Vektorski podatci**: Prostorni objekti su predstavljeni kao točke (lokacije vrsta), linije (rijeke, ceste) ili poligoni (šume, zaštićena područja).
3. **Prostorna rezolucija**: Određuje najmanji prostorni element koji može biti razlučen u analizi. Viša rezolucija znači manje ćelije i detaljniju informaciju, ali i veće računalne zahtjeve.

Matematički okvir prostornih podataka:

Za rasterske podatke, prostorna lokacija (i, j) u mreži odgovara geografskim koordinatama (x, y):

$$x = x_0 + i \times \Delta x \tag{6.1}$$

$$y = y_0 + j \times \Delta y \tag{6.2}$$

gdje su x_0 i y_0 koordinate ishodišta mreže, a Δx i Δy su prostorne rezolucije u x i y smjeru.

6.1.2 Daljinska istraživanja i satelitski podatci

Daljinska istraživanja pružaju kontinuirane vremenske serije podataka o stanju Zemljine površine, što je ključno za praćenje promjena u ekološkim sustavima.

Glavni izvori satelitskih podataka:

1. **Landsat serija**: Prostorna rezolucija 15-120 m, vremenske serije od 1972. godine
2. **MODIS (Moderate Resolution Imaging Spectroradiometer)**: Dnevni podatci globalne pokrivenosti, 250-1000 m rezolucija
3. **Sentinel-2**: 10-60 m rezolucija, 5-dnevni ciklus ponovnog snimanja

Vegetacijski indeksi:
Najčešće korišten je Normalizirani vegetacijski indeks (NDVI):

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}} \tag{6.3}$$

gdje su NIR (near-infrared) i Red reflektance vrijednosti u odgovarajućim spektralskim pojasevima. NDVI vrijednosti kreću se od -1 do +1, pri čemu više vrijednosti označavaju gušću i zdraviju vegetaciju.

6.1.3 Prostorne skale u ekologiji

Konceptualna razina analize ključna je za razumijevanje ekoloških procesa jer se različiti procesi manifestiraju na različitim prostornim skalama.

Lokalna skala (<1 km)

Na lokalnoj skali dominiraju mikroklima i mikrostanišni uvjeti. Ova skala je relevantna za:

- **Populacijsku dinamiku** malih organizama
- **Međuspecijske interakcije** (konkurencija, predacija)
- **Stanišne preferencije** pojedinačnih vrsta

Matematički pristup lokalnoj skali:
Za modeliranje na lokalnoj skali često koristimo funkcije kernela za opisivanje prostornih interakcija:

$$K(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{6.4}$$

gdje je d udaljenost između lokacija, a σ parametar koji kontrolira prostorni doseg interakcije.

Krajobrazna skala (1-100 km)

Krajobrazna skala obuhvaća mozaik različitih staništa i tipova korištenja zemljišta. Na ovoj skali proučavamo:

- **Fragmentaciju staništa**
- **Koridore i ekološku povezanost**
- **Metapopulacijske dinamike**

Metrike krajobrazne strukture:
1. **Indeks fragmentacije**:

$$\text{FI} = 1 - \frac{A_{\text{core}}}{A_{\text{total}}} \tag{6.5}$$

gdje je A_{core} površina osnovnih staništa, a A_{total} ukupna površina staništa.

2. **Indeks povezanosti**:

$$\mathrm{PC} = \frac{\sum_i \sum_j (a_i \times a_j \times p^*_{ij})}{A_L^2} \tag{6.6}$$

gdje su a_i i a_j površine fragmenata i i j, p^*_{ij} je vjerojatnost disperzije između fragmenata, a A_L je ukupna površina krajolika.

Regionalna skala (100-1000 km)

Regionalna skala odgovara biogeografskim regijama i karakterizirana je:

- **Klimatskim gradijentima**
- **Historijskim biogeografskim procesima**
- **Velikim ekološkim koridorima**

Kontinentalna skala (>1000 km)

Na kontinentalnoj skali dominiraju:

- **Makroklimatski uvjeti**
- **Biogeografske barijere**
- **Evolucijski procesi**

6.1.4 Prostorna autokorelacija

Fundamentalni princip prostorne ekologije glasi da su lokacije koje su bliže jedna drugoj sličnije od onih koje su udaljenije (Toblerovo prvo pravilo geografije).

Moranovo I:

Globalna prostorna autokorelacija mjeri se Moranovim I indeksom:

$$I = \frac{n}{S_0} \times \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{6.7}$$

gdje su:

- n broj lokacija
- w_{ij} prostorni težinski elementi
- $S_0 = \sum_i \sum_j w_{ij}$
- x_i vrijednost varijable na lokaciji i
- $\bar{x}$ prosjek varijable

Vrijednosti I kreću se od -1 (savršena negativna autokorelacija) do +1 (savršena pozitivna autokorelacija).

6.2 Modeli rasprostiranja vrsta (SDM/ENM)

Modeli rasprostiranja vrsta (Species Distribution Models - SDM) ili modeli ekoloških niša (Ecological Niche Models - ENM) predstavljaju jedan od najvažnijih alata u suvremenoj ekologiji i biologiji konzervacije.

6.2.1 Konceptualni okvir

Fundamentalni vs. realizirani ekološki prostor:
Hutchinsonov koncept ekološke niše razlikuje:

1. **Fundamentalna niša** (n-dimenzionalni hipervolumen): svi uvjeti okoliša u kojima vrsta može preživjeti i razmnožavati se
2. **Realizirana niša**: dio fundamentalne niše koje vrsta stvarno nastanjuje

Matematički zapis niše:
Ako je okoliš opisan s m varijabli $\mathbf{E} = (E_1, E_2, \ldots, E_m)$, onda je fundamentalna niša:

$$N_f = \{\mathbf{E} \in \mathbb{R}^m \mid \text{fitness}(\mathbf{E}) > 0\} \tag{6.8}$$

a realizirana niša:

$$N_r = N_f \cap A \cap B \tag{6.9}$$

gdje A predstavlja geografski dostupan prostor, a B biotičke interakcije.

6.2.2 Klimatske varijable i bioklimatski slojevi

WorldClim bioklimatske varijable:
Standardni skup od 19 bioklimatskih varijabli (BIO1-BIO19) karakterizira temperaturne i oborinske uvjete:

Temperaturne varijable:

- BIO1: Godišnja srednja temperatura
- BIO2: Srednji dnevni raspon temperatura (mjesečni prosjek(max temp - min temp))
- BIO3: Izotermnost (BIO2/BIO7) $\times$ 100
- BIO4: Temperaturna sezonalnost (standardna devijacija $\times$ 100)
- BIO5: Maksimalna temperatura najtopilijeg mjeseca
- BIO6: Minimalna temperatura najhladnijeg mjeseca
- BIO7: Godišnji temperaturni raspon (BIO5-BIO6)

Oborinske varijable:

- BIO12: Godišnji oborinski ukupak
- BIO13: Oborinski ukupak najkišnijeg mjeseca
- BIO14: Oborinski ukupak najsušljeg mjeseca
- BIO15: Oborinska sezonalnost (koeficijent varijacije)

Kombinacijske varijable:

- BIO16: Oborinski ukupak najkišnije četvrtine
- BIO17: Oborinski ukupak najsušlje četvrtine
- BIO18: Oborinski ukupak najtopiije četvrtine
- BIO19: Oborinski ukupak najhladnije četvrtine

Izračun sezonalnosti:
Temperaturna sezonalnost:

$$\text{BIO4} = \sigma(T_1, T_2, \ldots, T_{12}) \times 100 \tag{6.10}$$

Oborinska sezonalnost:

$$\text{BIO15} = \frac{\sigma(P_1, P_2, \ldots, P_{12})}{\mu(P_1, P_2, \ldots, P_{12})} \times 100 \tag{6.11}$$

gdje su T_i i P_i mjesečne temperature i oborini.

6.2.3 Algoritmi SDM modeliranja

Maximum Entropy (MaxEnt)

MaxEnt je jedan od najšire korištenih algoritma za SDM. Temelji se na principu maksimalne entropije - pronalaženje distribucije koja je najpristranija prema dostupnim podacima.

Matematički okvir MaxEnt:
Za skup okolišnih varijabli $\mathbf{X} = (x_1, x_2, \ldots, x_m)$ i skup značajki $\mathbf{f} = (f_1, f_2, \ldots, f_k)$, MaxEnt traži distribuciju q koja maksimizira entropiju:

$$H(q) = -\int q(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x} \tag{6.12}$$

uz ograničenja:

$$E_q[f_j] = \hat{f}_j \quad (j = 1, \ldots, k) \tag{6.13}$$

gdje je $\hat{f}_j$ empirijski prosjek značajke f_j iz podataka o prisutnosti.

Exponential family format:
Rješenje MaxEnt problema ima oblik:

$$q(\mathbf{x}) = \frac{\exp\left(\sum_j \lambda_j f_j(\mathbf{x})\right)}{Z} \tag{6.14}$$

gdje su λ_j Lagrangeovi multiplikatori, a Z normalizacijska konstanta:

$$Z = \int \exp\left(\sum_j \lambda_j f_j(\mathbf{x})\right) d\mathbf{x} \tag{6.15}$$

Logistički format:
Za praktičnu primjenu, MaxEnt koristi logistički format:

$$P(\mathbf{x}) = \frac{\exp\left(\sum_j \lambda_j f_j(\mathbf{x})\right)}{1 + \exp\left(\sum_j \lambda_j f_j(\mathbf{x})\right)} \tag{6.16}$$

Generalizirani linearni modeli (GLM)

GLM pristup koristi logističku regresiju za modeliranje prisutnosti/odsutnosti:

$$\text{logit}(p) = \beta_0 + \sum_i \beta_i x_i \tag{6.17}$$

gdje je p vjerojatnost prisutnosti vrste, a x_i su okolišne varijable.

Likelihood funkcija:

Za n lokacija s poznatim statusom prisutnosti $y_i \in \{0, 1\}$:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} p(\mathbf{x}_i)^{y_i}(1 - p(\mathbf{x}_i))^{1-y_i} \tag{6.18}$$

Random Forest

Random Forest kombinira mnoge stabla odlučivanja:

$$\hat{P}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x}) \tag{6.19}$$

gdje je B broj stabala, a $T_b(\mathbf{x})$ predviđanje b-tog stabla.

6.2.4 Evaluacija modela

AUC (Area Under the ROC Curve):

ROC krivulja prikazuje osjetljivost (True Positive Rate) u odnosu na 1-specifičnost (False Positive Rate):

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.20}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{6.21}$$

AUC vrijednosti:

- 0.5: random model
- 0.7-0.8: dobar model
- 0.8-0.9: vrlo dobar model
- >0.9: izvrsno (možda overfitting)

TSS (True Skill Statistic):

$$\text{TSS} = \text{Sensitivity} + \text{Specificity} - 1 \tag{6.22}$$

TSS kreće se od -1 do +1, gdje vrijednosti >0.4 označavaju dobre modele.

6.2.5 Pristranost uzorkovanja

Pristranost uzorkovanja predstavlja jedan od najvećih izazova u SDM modeliranju.

Prostorna pristranost:

Opažanja su često koncentrirana uz ceste, naselja ili dostupna područja. Ovo može dovesti do:

1. **Geografske pristranosti**: područja nisu jednoliko uzorkovana
2. **Okolišne pristranosti**: određeni tip staništa je predviđen zbog dostupnosti

Metode korekcije pristranosti:

1. **Spatial filtering**: uklanjanje blisko lokaliziranih točaka

$$\text{min\_distance} = \sigma \times \sqrt{\frac{A}{n}} \tag{6.23}$$

gdje je A ukupna površina studijskog područja, n broj lokacija, σ scaling faktor.

2. **Target-group background**: korištenje lokacija drugih vrsta kao pozadinske točke

3. **Kernel density estimation**: težinski background uzorčavanje prema distribuciji opažanja

$$w(\mathbf{x}) = \frac{K(\|\mathbf{x} - \mathbf{x}_i\|/h)}{\sum_j K(\|\mathbf{x} - \mathbf{x}_j\|/h)} \tag{6.24}$$

gdje je K kernel funkcija (obično Gaussian), h bandwidth parametar.

6.3 Individualno temeljeni modeli (IBM)

Individualno temeljeni modeli (Individual-Based Models - IBM) predstavljaju bottom-up pristup modeliranju gdje se ponašanje sustava izvodi iz interakcija između jedinki.

6.3.1 Konceptualni okvir IBM-a

Osnovni principi:

1. **Diskretnost**: jedinke su diskretni entiteti s vlastitim svojstvima
2. **Lokalnost**: interakcije se odvijaju lokalno u prostoru i vremenu
3. **Emergentno**: svojstva populacije i zajednice nastaju iz interakcija među jedinkama

Struktura IBM modela:
Svaka jedinka $i \in \{1, 2, \ldots, N\}$ karakterizirana je skupom atributa:

$$I_i = (x_i, y_i, a_i, s_i, e_i, \ldots) \tag{6.25}$$

gdje su:

- (x_i, y_i): prostorna lokacija
- a_i: dob
- s_i: spol
- e_i: energetsko stanje

6.3.2 Prostorno ponašanje i movement

Random walk modeli:
Jednostavan random walk u diskretnom vremenu:

$$x_i(t+1) = x_i(t) + \Delta x \tag{6.26}$$

$$y_i(t+1) = y_i(t) + \Delta y \tag{6.27}$$

gdje su Δx i Δy slučajni pomaci iz određene distribucije.
Correlated random walk:

$$\theta_i(t+1) = \theta_i(t) + \delta \tag{6.28}$$

gdje je θ smjer kretanja, a $\delta \sim N(0, \sigma^2)$ slučajna promjena smjera.
Biased random walk:
Uključuje preferenciju prema određenim staništima:

$$P(\text{move to cell } j) \propto \exp(\beta S_j) \tag{6.29}$$

gdje je S_j prikladnost staništa u ćeliji j.

6.3.3 Energetski modeli

Simple energy budget:

$$\frac{dE}{dt} = I - M - G \tag{6.30}$$

gdje je:

- E: energetski sadržaj jedinke
- I: stopa unosa energije (hranjenje)
- M: stopa metabolizma
- G: stopa rasta/reprodukcije

Foraging behaviour:
Optimalno hranjenje prema Marginal Value Theorem:

$$\frac{\partial G}{\partial t} = \lambda \tag{6.31}$$

gdje je λ prosječna stopa dobivanja energije u staništu.
Functional response:
Type II functional response za individual foraging:

$$I = \frac{a \times R}{1 + a \times h \times R} \tag{6.32}$$

gdje je:

- a: attacking rate
- R: resource density
- h: handling time

6.3.4 Reprodukcija i demografija

Probabilistic reproduction:
Vjerojatnost reprodukcije ovisi o energetskom stanju:

$$P(\text{reproduction}) = \frac{1}{1 + \exp(-\beta(E - E_{\text{threshold}}))} \tag{6.33}$$

Offspring numbers:
Broj potomaka slijedi Poisson distribuciju:

$$N_{\text{offspring}} \sim \text{Poisson}(\lambda_{\text{reproduction}} \times \text{fecundity}) \tag{6.34}$$

6.3.5 Interakcije između jedinki

Competition for resources:
Lokalna konkurencija modelirana kernel funkcijom:

$$\text{Competition\_effect}_i = \sum_{j \neq i} K(d_{ij}) \times \text{resource\_overlap} \tag{6.35}$$

gdje je $K(d) = \exp(-d^2/2\sigma^2)$ competition kernel.
Predator-prey interactions:
Vjerojatnost predacije:

$$P(\text{predation}) = 1 - \exp(-\lambda \times \text{predator\_density} \times \text{exposure\_time}) \tag{6.36}$$

6.3.6 Emergentna svojstva

IBM modeli mogu proizvesti kompleksna emergentna ponašanja:

Spatial aggregation:
Koeficijent agregacije:

$$IA = \frac{\sigma^2/\mu - 1}{n - 1} \tag{6.37}$$

gdje je σ^2 varijanca, μ prosjek gustoće, n broj ćelija.

Population cycles:
Analizom vremenskih serija ukupne abundancije:

$$N(t) = A + B\cos\left(\frac{2\pi t}{T} + \phi\right) \tag{6.38}$$

gdje je T period ciklusa.

Pattern formation:
Prostorni uzorci analizirani prostornim autokorelacijskim funkcijama:

$$r(d) = \frac{\sum_i \sum_j (n_i - \bar{n})(n_j - \bar{n})}{\sigma^2 \times \text{number\_of\_pairs}} \tag{6.39}$$

6.3.7 Implementacija i softverski alati

NetLogo pseudokod:

```
to setup
create-turtles initial-population [
setxy random-xcor random-ycor
set energy random 100
set age 0
]
end

to go
ask turtles [
move
feed
reproduce
age
if energy < 0 [ die ]
]
tick
end

to move
rt random 360
fd step-size
end
```

Kalibracija i validacija:
IBM modeli kalibriraju se pomoću:

1. **Pattern-oriented modelling (POM)**: fitiranje multiplih prostornih i temporalnih uzoraka
2. **Approximate Bayesian Computation (ABC)**:

$$d(S_{\text{obs}}, S_{\text{sim}}) < \varepsilon \tag{6.40}$$

   gdje je d distance funkcija između opaženih (S_{obs}) i simuliranih (S_{sim}) uzoraka.

IBM modeli predstavljaju moćan alat za razumijevanje kompleksnih ekoloških sustava, omogućujući nam da istražimo kako lokalne interakcije generiraju globalne obrasce u prostoru i vremenu.

Poglavlje 7

Prediktivno modeliranje i scenariji

7.1 Scenariji klimatskih promjena

Korištenje scenarija (npr. SSP/RCP) u ekološkim projekcijama.

7.2 Predviđanje invazija i rani sustavi upozorenja

Integracija podataka nadzora i modela rizika; pragovi upozorenja.

7.3 Spajanje mehanističkih i statističkih pristupa

Hibridni modeli i kalibracija.

Poglavlje 8

Statistički i AI pristupi

8.1 Klasična i Bayesovska statistika

GLM/GLMM, MCMC, kredibilni intervali.

8.2 Strojno učenje

RF, XGBoost, SVM; značajke, regularizacija, unakrsna provjera.

8.3 Duboko učenje i digitalni blizanci ekosustava

Grafovi, konvolucijske i sekvencijske mreže; digital twins za *what-if* scenarije.

Dio IV

Validacija, nesigurnost i primjene

Poglavlje 9

Validacija, verifikacija i osjetljivost

9.1 Kalibracija i verifikacija

Podjela podataka, *out-of-sample* procjene, nezavisni skupovi.

9.2 Analize osjetljivosti i robustnosti

Lokalne i globalne metode; Monte Carlo; propagacija nesigurnosti.

9.3 Etika i komunikacija nesigurnosti

Transparentnost i odgovornost modelara.

Poglavlje 10

Primjene u upravljanju okolišem

10.1 Zaštita prirode i bioraznolikosti

Prioritizacija očuvanja, planovi upravljanja.

10.2 Poljoprivreda i šumarstvo

Produktivnost, štetnici, otpornost agroekosustava.

10.3 Ekotoksikologija i procjena rizika

LC/EC metri
ke, PNEC, scenariji izloženosti; integracija s populacijskim modelima.

10.4 Klimatske politike i održivi razvoj

Podrška odlučivanju, socio-ekonomske poveznice.

Dio V

Studije slučaja i praktične vježbe

Poglavlje 11

Studije slučaja

11.1 Gujavice u agroekosustavima

Dobno strukturirani modeli, elastičnost i scenariji upravljanja.

11.2 Širenje komaraca

SDM + meteorološki pogonjeni prediktori; rani sustavi upozorenja.

11.3 Eutrofikacija riječnog ekosustava

Kutije (box) modeli i validacija na nizvodnim mjerenjima.

11.4 Fragmentacija šuma i ptice metapopulacija

Povezanost staništa i pragovi propusnosti krajolika.

11.5 Bayesove mreže u ekotoksikologiji

Kauzalni grafikoni i inverzna inferencija.

Poglavlje 12

Praktične vježbe (R i Python)

12.1 Uvod u R i Python alate

Instalacija, radna okolina, reproducibilnost.

12.2 Leslie matrica u R

Listing 12.1: Leslie matrica i projekcija populacije u R-u

```
F <- c(0, 0.3, 1.2, 1.5)
P <- c(0.6, 0.7, 0.8)
L <- matrix(c(F,
P[1],0,0,0,
0,P[2],0,0,
0,0,P[3],0), nrow=4, byrow=TRUE)
n0 <- c(50, 40, 20, 10)
proj <- function(L, n, t=20){
       N <- matrix(NA, nrow=length(n), ncol=t+1)
       N[,1] <- n
       for(i in 1:t) N[,i+1] <- L %*% N[,i]
       N
}
N <- proj(L, n0, t=30)
colSums(N) -> Tot
plot(Tot, type="l", xlab="Vrijeme", ylab="Uk. čveliina")
```

12.3 Lotka–Volterra simulacija u Python

Listing 12.2: Jednostavni LV model u Pythonu (scipy.integrate)

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

def lv(t, z, r, a, e, m):
N, P = z
dN = r*N - a*N*P
dP = e*a*N*P - m*P
return [dN, dP]
```

```
pars = dict(r=0.8, a=0.02, e=0.1, m=0.3)
sol = solve_ivp(lambda t,z: lv(t,z,**pars),
[0, 200], [40, 9], dense_output=True)
t = np.linspace(0,200,1000)
N, P = sol.sol(t)
plt.plot(t, N, label="Plijen")
plt.plot(t, P, label="žGrabeljivac")
plt.xlabel("Vrijeme"); plt.ylabel("ćGustoa"); plt.legend(); plt.show
    ()
```

12.4 SDM u R (skica s dismo)

Listing 12.3: SDM skica s bioklimatskim varijablama

```
library(dismo); library(raster)
# bioclim <- getData('worldclim', var='bio', res=10) # primjer
    ćdohvaanja
# occ <- read.csv("occ_points.csv") # popisi žopaanja (lon, lat)
# m <- maxent(bioclim, occ)
# p <- predict(bioclim, m)
# plot(p)
```

12.5 Random Forest za invazije

Listing 12.4: RF klasifikator s unakrsnom provjerom

```
library(tidymodels)
set.seed(1)
# df: response ~ predictors
split <- initial_split(df, prop=0.8, strata=response)
train <- training(split); test <- testing(split)
rf_spec <- rand_forest(trees=500) %>% set_engine("ranger") %>% set_
    mode("classification")
rec <- recipe(response ~ ., data=train) %>% step_zv(all_predictors())
wf <- workflow() %>% add_model(rf_spec) %>% add_recipe(rec)
res <- wf %>% fit_resamples(vfold_cv(train, v=5, strata=response),
    metrics=metric_set(roc_auc,accuracy))
collect_metrics(res)
```

12.6 DEB simulacije (skica)

Listing 12.5: Minimalna DEB skica (konceptualno)

```
# Ovo je conceptual stub: definirajte parametre i tokove E, V...
pars <- list(p_Am=1, v=0.02, kappa=0.8)
state <- c(E=1, V=0.1)
deb_model <- function(t, y, p){
       with(as.list(c(y,p)), {
              dE <- p_Am - v*E
              dV <- kappa*v*E - 0.01*V
              list(c(dE,dV))
       })
```

```
        }
        # solve s deSolve::ode(...)
```

Dodatak A

Matematički prilozi

A.1 Osnovni populacijski modeli

Sažeti popis formula (eksponencijalni, logistički, Ricker, Beverton–Holt).

Dodatak B

Instalacija softvera

Upute za R/Python okruženja, pakete i reproducibilnost.

Dodatak C

Primjeri koda

Proširene skripte za poglavlja iz *Praktičnih vježbi*.

Dodatak D

Rječnik pojmova

Bioraznolikost, nosivost staništa, metapopulacija, ekološka niša, stohastičnost.