Optimizing Bioenergetic Food Web Models of Ecosystems Using Gamification and Machine Learning

Ben Saylor
Master’s Thesis Defense
San Francisco State University
May 15, 2017
Introduction

• Ecosystems are complex systems; ecological modeling is complex

• *Allometric trophic network (ATN)* models
  
  • Model population dynamics in terms of energy flow through *food webs*
  
  • Difficult to parameterize due to highly complex and nonlinear behavior

• Our objective: find ways of parameterizing ATN models to replicate sustaining ecosystems

• Our approaches: based on machine learning and gamification
Contributions

• Graph sampling algorithm for food webs
• New ATN simulator implementation
  • Improves correctness and performance over previous implementation
  • Adds steady state detection
• New environment score formula
• Showed that ML can predict ecosystem health
• Method of generating persistent simulated ecosystems
  • New ecosystems for Convergence game
• ML-based method for generating parameter hints
• ML-based method of searching ATN model parameter space
ATN models

- System of ordinary differential equations

\[ B'_i = f(B) \]
ATN model equations

Producers:

\[ B'_i = r_i B_i G_i(B) - \sum_{j \in \text{predators}} (x_j y_{ji} \alpha_{ji} F_{ji} B_j / e_{ji}) \]

Consumers:

\[ B'_i = \sum_{j \in \text{prey}} (x_i y_{ij} \alpha_{ij} F_{ij} B_i) - \sum_{j \in \text{predators}} (x_j y_{ji} \alpha_{ji} F_{ji} B_j / e_{ji}) - x_i B_i \]

Growth function:

\[ G_i(B) = 1 - \frac{B_i}{K_i} \]

Functional response:

\[ F_{ij} = \frac{B_j^{1+q_{ij}}}{\sum_{m \in \text{prey}} \alpha_{im} B_m^{1+q_{im}} + B_{0ij}^{1+q_{ij}}} \]
## ATN model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{ij}$</td>
<td>Relative half-saturation density of predator $i$ when consuming prey $j$</td>
</tr>
<tr>
<td>$B_{0ij}$</td>
<td>Half-saturation density of predator $i$ when consuming prey $j$</td>
</tr>
<tr>
<td>$c_{ij}$</td>
<td>Competition between producers $i$ and $j$ for shared carrying capacity</td>
</tr>
<tr>
<td>$e_{ij}$</td>
<td>Assimilation efficiency of prey $j$ by predator $i$</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Carrying capacity of producer $i$</td>
</tr>
<tr>
<td>$K_s$</td>
<td>System-wide carrying capacity</td>
</tr>
<tr>
<td>$q_{ij}$</td>
<td>Functional response control parameter</td>
</tr>
<tr>
<td>$r_i$</td>
<td>Growth rate of producer $i$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Metabolic rate of consumer $i$</td>
</tr>
<tr>
<td>$y_{ij}$</td>
<td>Maximum ingestion rate of predator $i$ when consuming prey $j$</td>
</tr>
</tbody>
</table>
ATN model simulation

• Integrate the equations, solving the *initial value problem*

• Result: biomass over time by species
Gamification

- Crowdsourcing human problem-solving abilities to help solve problems that are hard using computation alone
- Successful example: FoldIt
- *World of Balance* by Dr. Yoon et al.: ecosystem nurturing game backed by ATN simulations
- Mini-games within WoB include *Convergence*
Convergence game

Select Ecosystem:
- Ecosystem #1 (5 species)
- Ecosystem #2 (6 species)
- Ecosystem #3 (6 species)
- Ecosystem #4 (5 species)
- Ecosystem #3 (11 species)

Convergence

Attempt #2

Target Graph

Biomass

Month

African Clawless Otter [70] - X
Tree Mouse [31] - X
Crickets [14] - X
Grass And Herbs [5] - X

Submit Progress Report Get Hint
Reset to: Initial #1 #2 #3
Food webs as graphs

- Food webs are directed graphs
- Edges point from prey to predator

<table>
<thead>
<tr>
<th>Graph concept</th>
<th>Food web concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>species, guild, resource</td>
</tr>
<tr>
<td>edge</td>
<td>feeding relationship</td>
</tr>
<tr>
<td>self-loop</td>
<td>cannibalism</td>
</tr>
<tr>
<td>source node</td>
<td>producer</td>
</tr>
<tr>
<td>non-source node</td>
<td>consumer</td>
</tr>
<tr>
<td>directed simple path from source node</td>
<td>food chain</td>
</tr>
</tbody>
</table>
Graph sampling and “subwebs”

87 Serengeti species

8 Serengeti species

Motivations: analysis, gameplay
Subweb criteria

1. Single connected component
2. No incomplete food chains
3. Contains all edges between selected nodes
Random subweb algorithm

- Start with given number of producer nodes
- Each iteration, grow outward:
  - Try adding a predator of a “lonely plant eater”
  - If none, add any neighboring consumer
- Add all edges between selected nodes
Random subweb algorithm
2 Plant Juices
3 Fruits and Nectar
4 Grains and Seeds
5 Grass and Herbs
6 Trees and Shrubs
15 Herbivorous True Bugs
16 Katydid
26 Yellow-Breasted Apalis
27 Yellow-Bellied Eremomela
31 Tree Mouse
33 Cape Teal
41 Striped Weasel
43 Coqui Francolin
49 African Marsh Owl
50 Dwarf Mongoose
51 Hooded Vulture
53 African Fish Eagle
55 Greater Bushbaby
61 Leopard Tortoise
64 Nile Monitor Lizard
65 Kori Buskard
67 Black Backed Jackal
69 Serval Cat
71 Kirk's Dik-dik
72 Crested Porcupine
73 Oribi
75 Black Mamba
80 Leopard
85 Ostrich
86 Lion
89 Southern Eland
91 Nile Crocodile
94 Hippopotamus
95 African Elephant
Measuring ecosystem health

- Goals:
  - Provide WoB players with feedback via score
  - Provide target variables for machine learning
- WoB Environment Score
  - Rewards high biomass levels
  - Rewards presence of high trophic level species
  - Rewards species diversity
Original Environment Score

\[
\left[ 5 \log_2 \left( \sum_{i=1}^{N} b_i \left( \frac{B_i}{b_i} \right)^{T_i} \right) \right]^2 + N^2
\]

- Potential disadvantages:
  - Top predators and small animals can contribute orders of magnitude more weight
  - WoB body size data is estimated
  - \( N^2 \) does not scale with biomass - species diversity makes a very small contribution
Revised Environment Score

\[
\text{BiomassScore} = \sum_{i=1}^{N} T_i B_i
\]

\[
\text{Shannon} = -\sum_{i=1}^{N} p_i \log_2 p_i \quad \text{where } p_i = \frac{B_i}{\sum_{j=1}^{N} B_j}
\]

\[
\text{RevisedEnvironmentScore} = \text{BiomassScore} \times (1 + \text{Shannon})
\]

• Advantages:
  • Species contributions are more balanced
  • Species diversity makes a larger contribution
Parameter range hints for Convergence

• Goal: guide Convergence players by showing promising parameter ranges

• 4-step approach:

  1. **Species selection**: use our graph sampling algorithm to generate food webs

  2. **Parameter space exploration and simulation**: Generate 1,000s of simulations with randomized parameters, evaluate ecosystem health using linear time trend of environment score

  3. **Machine-learning classification of simulation results**: Classify simulations as “good” or “bad” based on score trend values, train decision trees to predict label based on model parameters

  4. **Derivation of parameter ranges to display as game hints**: Use the decision tree structure to derive promising parameter ranges
Example of decision tree

X28 \leq 0.601404
  |   X51 \leq 0.14091
  |   |   X28 \leq 0.455708: good (11.0)
  |   |   X28 > 0.455708: bad (10.0/1.0)
  |   X51 > 0.14091
  |   |   X73 \leq 0.079958
  |   |   |   X86 \leq 0.069814: good (22.0)
  |   |   |   X86 > 0.069814: bad (5.0/1.0)
  |   |   X73 > 0.079958: good (157.0)
X28 > 0.601404
  |   X51 \leq 0.18639: bad (194.0)
  |   X51 > 0.18639
  |   |   K3 \leq 6003.88: good (56.0/1.0)
  |   |   K3 > 6003.88
  |   |   |   X73 \leq 0.145747: bad (41.0)
  |   |   |   X73 > 0.145747: good (4.0/1.0)
Example of derived ranges
### Evaluation: classifier performance

- Evaluated approach on 3 food webs

<table>
<thead>
<tr>
<th>Food web</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 species</td>
<td>good</td>
<td>0.951</td>
<td>0.951</td>
<td>0.951</td>
</tr>
<tr>
<td></td>
<td>bad</td>
<td>0.959</td>
<td>0.959</td>
<td>0.959</td>
</tr>
<tr>
<td>10 species</td>
<td>good</td>
<td>1.000</td>
<td>0.996</td>
<td>0.998</td>
</tr>
<tr>
<td></td>
<td>bad</td>
<td>0.996</td>
<td>1.000</td>
<td>0.998</td>
</tr>
<tr>
<td>15 species</td>
<td>good</td>
<td>0.996</td>
<td>1.000</td>
<td>0.998</td>
</tr>
<tr>
<td></td>
<td>bad</td>
<td>1.000</td>
<td>0.996</td>
<td>0.998</td>
</tr>
</tbody>
</table>
Evaluation: parameter ranges

- Conducted “simulated user study” with a test group given hints, a control group not given hints

<table>
<thead>
<tr>
<th>Food web</th>
<th>Statistic</th>
<th>Control group</th>
<th>Test group</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 species</td>
<td>mean</td>
<td>-3.054789</td>
<td>5.358788</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>9.410075</td>
<td>2.236706</td>
</tr>
<tr>
<td>10 species</td>
<td>mean</td>
<td>-5.187572</td>
<td>-6.016333</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>8.226043</td>
<td>2.130220</td>
</tr>
<tr>
<td>15 species</td>
<td>mean</td>
<td>-2.250668</td>
<td>-0.975169</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>5.101375</td>
<td>4.270391</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Food web</th>
<th>t-statistic</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 species</td>
<td>27.508</td>
<td>$1.772 \times 10^{-127}$</td>
</tr>
<tr>
<td>10 species</td>
<td>-3.0842</td>
<td>0.00209</td>
</tr>
<tr>
<td>15 species</td>
<td>6.0628</td>
<td>$1.6038 \times 10^{-9}$</td>
</tr>
</tbody>
</table>
Steady states
Steady states: Implementation

• Basis: ATN model is memoryless: $B_t' = f(B_t)$

• Watch for zero derivatives: \[ \left| \frac{B_i'}{B_i} \right| \leq 10^{-10} \]

• Watch for return to previous state: \[ \left| \frac{B_i - B_{si}}{B_{si}} \right| \leq 0.01 \]

• Wait for this to occur 3 times for confirmation
## Steady states: Evaluation

<table>
<thead>
<tr>
<th>Food web</th>
<th>q</th>
<th>TP</th>
<th>FP</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-8-9-26-41</td>
<td>0.0</td>
<td>319</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>954</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td>3-21-55-80-85</td>
<td>0.0</td>
<td>29</td>
<td>1</td>
<td>0.967</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>432</td>
<td>86</td>
<td>0.834</td>
</tr>
<tr>
<td>3-30-50-69-71</td>
<td>0.0</td>
<td>17</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>929</td>
<td>50</td>
<td>0.949</td>
</tr>
<tr>
<td>2-3-5-8-9-21-22-69-71-94</td>
<td>0.0</td>
<td>718</td>
<td>1</td>
<td>0.999</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>725</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td>4-7-14-43-47-61-69-74-80-89</td>
<td>0.0</td>
<td>836</td>
<td>0</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>850</td>
<td>0</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Generating sustaining ecosystems for Convergence

• Objective: generate simulated ecosystems in which
  • all species survive, and
  • biomass graph is visually interpretable.

• Approach:
  1. Run many simulations to steady state including animals
  2. Drop species that went extinct, assemble food web from remaining species
  3. Generate new simulations starting with final biomass from previous simulations, pruned food webs, same parameters
  4. Automatically filter results for visual clarity
Results

• Preliminary experiment filtered 4,000 simulations down to 384 such as the one below
Using decision trees to narrow the parameter search space

- Objective: For a given food web, identify regions of the parameter space leading to sustaining ecosystems
- Definition of “sustaining”: all species survive to steady state
- Approach: iteratively refine search space based on promising regions identified using decision trees
Decision tree search process

• Each iteration:
  
  • Generate 1,000 simulations with parameter values randomly drawn from the promising regions identified in the previous iteration
  
  • Train and test a decision tree to classify “good” vs. “bad” simulations based on median extinction count
  
  • Identify “promising” decision tree leaves that predict “good” simulations
  
  • Follow the path to each leaf to obtain promising region bounds
Decision tree search results
Progress in reducing extinctions

![Graph showing progress in reducing extinctions for different numbers of species (5, 10, 15)].

- **Mean Extinction Count** against **Iteration**
- **Legend**:
  - Red line: 5 species
  - Blue line: 10 species
  - Purple line: 15 species
Classifier prediction performance

![Graph showing F1 score class average over iterations for 5, 10, and 15 species.]
Future work

• Convergence hints: user study, dynamic version

• Steady state detection improvement

• Decision tree search improvements: class balance, classification accuracy, computational performance

• Study persistent chaotic dynamics

• Consider system-wide K for WoB and Convergence

• Machine learning approaches that generalize across different food webs
Immediate future work

• Evaluation of effects of an alternative producer growth function on steady state detection

• Evaluation of using promising regions from decision tree search to generate Convergence ecosystems