calculating_extinction_probabilities

December 8, 2019

1 Determining extinction probabilities

Three of the IUCN categories by definition have a specified risk of extinction, which is commonly used in the literature (Kindvall & Gärdenfors, 2003; Redding & Mooers, 2006):

- VU = 10% chance to go extinct within the next 100 years
- EN = 20% chance to go extinct within the next 20 years or 5 generations, whichever is longer (max = 100 years)
- CR = 50% chance to go extinct within the next 10 years or 3 generations, whichever is longer (max = 100 years)

Strictly speaking, these extinction probabilities only apply for taxa assessed under criterion E and not for taxa assessed by criteria A-D, but here we assume that all taxa are assessed under criterion E. This is a major simplification but is necessary in order to retrieve extinction probabilities for each status. First we will interpolate yearly extinction probabilities for the statuses VU and EN, based on the "default" extinction risk for the categories EN and CR (not accoutning for generation time). Then we will calculate species-specific extinciton probabilities for statuses EN and CR, based on generation length estimates.

In order to extrapolate these probabilities to other than the defined time frames, Kindvall & Gärdenfors (2003) introduce a formula to calculate the extinciton risk per year (E) for these categories:

$$E_1 = 1 - \sqrt[t]{1 - E_t}$$

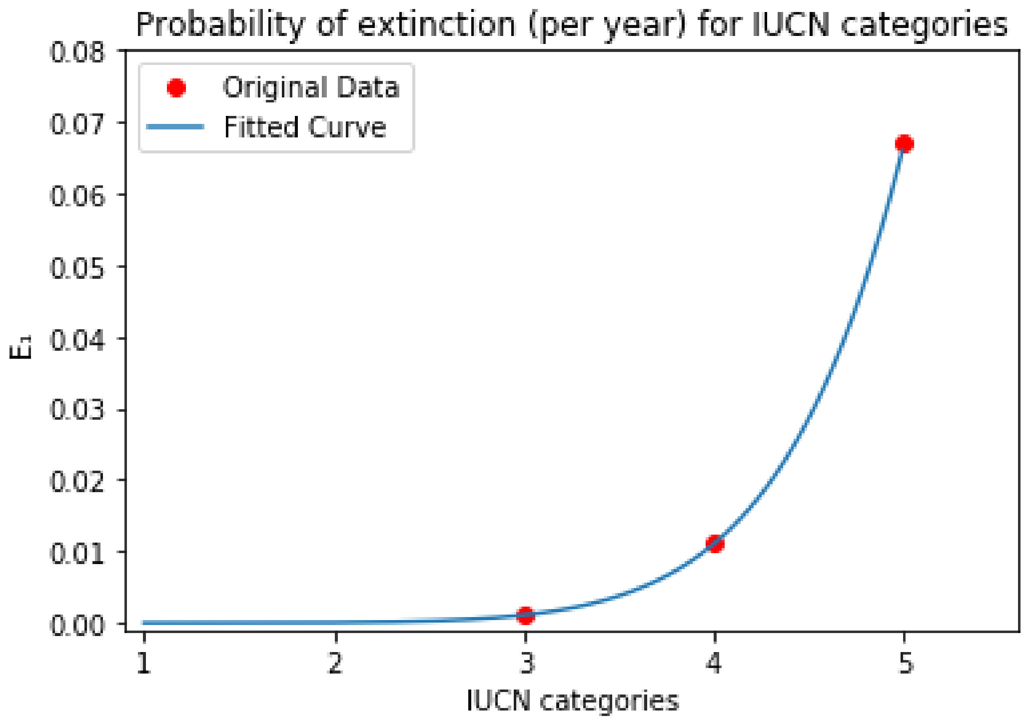
Applying this formula to the three categories mentioned above, which we have information for, returns:

[1]:
```python
def p_e_year(years,p_e):
    pe_year = 1-(1-float(p_e))**(1/float(years))
    return pe_year
p_year_VU = p_e_year(100,0.1)
p_year_EN = p_e_year(20,0.2)
p_year_CR = p_e_year(10,0.5)

print('E1_VU =', p_year_VU, '\nE1_EN =', p_year_EN, '\nE1_CR =', p_year_CR)
```

```
E1_VU = 0.0010530503095456112
E1_EN = 0.011095167094968383
E1_CR = 0.06696700846319259
```

Now, we want to fit a power function to these points in order to determine the values for categories 1 (=LC) and 2 (=NT):

[2]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
# setting the numpy parameters (printing floats in non-scientific notation and␣
 ↪with 12 decimal positions)
np.set_printoptions(suppress=True) # prints floats, no scientific notation
np.set_printoptions(precision=12) # rounds all array elements to 3rd digit

# Feeding in the data, consisting of the probabilities of extinction per year␣
 ↪for the categories 3 (=VU),4 (=EN) and 5 (=CR)
x = np.array([3.0,4.0,5.0])
y = np.array([0.0010530503095456112,0.011095167094968383,0.06696700846319259])

# defining the power function
def power_function(x,a,b):
    y = float(a)*x**float(b)
    return y

# fitting the power function to the 3 data points
a_b = curve_fit(power_function,x,y)

# extracting the values for a and b from the curve fit function
a = a_b[0][0]
b = a_b[0][1]

# create a series of numbers in order to plot the function
t = np.arange(1.0, 5.0, 0.01)

plt.plot(x, y, 'ro',label="Original Data")
plt.axis([0.9, 5.6, -0.001, 0.08])
plt.plot(t, power_function(t,a,b), label="Fitted Curve")
plt.legend(loc='upper left')
plt.xlabel('IUCN categories')
plt.ylabel('E ')
plt.title('Probability of extinction (per year) for IUCN categories')
plt.show()
```



We can use the estimated function to determine the values for categories 1 (=LC) and 2 (=NT):

[3]:
```
p_year_LC = power_function(1,a,b)
p_year_NT = power_function(2,a,b)
print('E1_LC =','{0:.12f}'.format(p_year_LC), '\nE1_NT =', '{0:.12f}'.
↪format(p_year_NT))
```

```
E1_LC = 0.000000155728
E1_NT = 0.000041551152
```

We can use formula 5 from Kindvall & Gärdenfors (2003):

$$E_t = 1 - (1 - E_1)^t$$

to extrapolate the risk of extinction for any category and time period t.

In order to get a better feeling of what these yearly probabilities mean and to check if our power function approximation returns sensible results, we apply this formula to the calculated yearly probabilities, using different time-frames for the different categories:

[4]:
```
def extrapolate_t_years(pe_year,time_period):
    result = 1-(1-float(pe_year))**float(time_period)
    return result

p_10000_LC = extrapolate_t_years(p_year_LC,10000)
p_1000_NT = extrapolate_t_years(p_year_NT,1000)
```

```
p_100_VU = extrapolate_t_years(p_year_VU,100)
p_20_EN = extrapolate_t_years(p_year_EN,20)
p_10_CR = extrapolate_t_years(p_year_CR,10)

print('Probability of LC taxon going extinct within 10000 years = {0:.2f}%'.
 ↪format(p_10000_LC*100))
print('Probability of NT taxon going extinct within 1000 years = {0:.0f}%'.
 ↪format(p_1000_NT*100))
print('Probability of VU taxon going extinct within 100 years = {0:.0f}%'.
 ↪format(p_100_VU*100))
print('Probability of EN taxon going extinct within 20 years = {0:.0f}%'.
 ↪format(p_20_EN*100))
print('Probability of CR taxon going extinct within 10 years = {0:.0f}%'.
 ↪format(p_10_CR*100))
```

```
Probability of LC taxon going extinct within 10000 years = 0.16%
Probability of NT taxon going extinct within 1000 years = 4%
Probability of VU taxon going extinct within 100 years = 10%
Probability of EN taxon going extinct within 20 years = 20%
Probability of CR taxon going extinct within 10 years = 50%
```

Redding & Mooers (2006) compare extrapolated extinction probabilities with real data and conclude that extrapolating extinction probabilities in this manner is generally fairly accurate, while it leads to an overestimation of the extinction risk for the most endangered category CR (5):

"For the EN category the p^e value was qualitatively very similar to the extrapolated p^e calculated by assuming extinction risk remains constant over 100 years (mean 0.315 vs. extrapolated 0.328). For CR the mean value of probability of extinction suggested that a lower score was more appropriate (mean 0.786 vs. extrapolated 0.999)."

1.0.1 Accounting for different generation times

In order to properly estimate the extinction risk for categories EN and CR, we need to take the generation time into account. Let's first read the generation length data for all taxa:

[5]:
```
import pandas as pd
gl_data = pd.read_csv('../../processed/generation_length/gl_all_mammals.
 ↪txt',sep='\t')
```

Now we calculate the yearly extinction risk for all species for the statuses EN and CR.

[6]:
```
species_names = [species.species for i,species in gl_data.iterrows()]
en_risks = [p_e_year(min(max(20,5*species.generation_length_days/365),100),0.2)␣
 ↪for i,species in gl_data.iterrows()]
cr_risks = [p_e_year(min(max(10,3*species.generation_length_days/365),100),0.5)␣
 ↪for i,species in gl_data.iterrows()]
```

We write the data to file for our future simulations.

[7]:
```
final_df = pd.DataFrame(data=np.array([species_names,en_risks,cr_risks]).
↪T,columns=['species','en_ex_risk','cr_ex_risk'])
final_df.to_csv('../../processed/iucn_data/en_cr_extinction_risks_all_species.
↪txt',sep='\t',index=False)
```

1.1 References

- Kindvall, O., & Gärdenfors, U. (2003). Temporal Extrapolation of PVA Results in Relation to the IUCN Red List Criterion E. Conservation Biology, 17(1), 316–321.
- Redding, D. W., & Mooers, A. O. (2006). Incorporating evolutionary measures into conservation prioritization. Conservation Biology, 20(6), 1670–1678. http://doi.org/10.1111/j.1523-1739.2006.00555.x