Christian Felt and Jadon Wagstaff

Data Mining Spring 2017

Instructor: Jeff Phillips

March 9, 2017

Intermediate Report

Christian Felt

Using ArcMap software, I computed the centroids of the polygons that represent the ranges of terrestrial species in the IUCN Endangered Species Red List shapefiles. Using my own Python code, I turned the latitude and longitude coordinates of these centroids into a 2-dimensional matrix and performed various kinds of clustering, trying to find areas of especially high density of endangered species.

To measure distance, I used the great-circle distance as given by d from the Haversine formula:

a = sin²(Δφ/2) + cos φ1 ⋅ cos φ2 ⋅ sin²(Δλ/2)

c = 2 ⋅ atan2( ,  )

d = R ⋅ c

Since the Earth is not perfectly spherical, the great-circle distance may be wrong by up to approximately 0.3% or 22km according to the analysis at <http://gis.stackexchange.com/q/25494>. In their article “Global Patterns of Terrestrial Vertebrate Diversity and Conservation,” Jenkins et al set 100km as a reasonable lower limit for “fine-grained spatial analysis” in this field.

Hierarchical clustering ran too slowly for the full dataset (74,529 points) or for amphibians alone (18,694 points). K-means++ worked for small numbers of clusters (20 or less) but became obnoxiously slow for larger k. Lloyd’s was also too slow for more than about 20 clusters.

Gonzalez clustering processed the entire dataset in less than 10 seconds. Using the “elbow” principle, I chose k=40 (see Figure 1). The resulting clusters (see Figure 2) seem surprisingly reasonable, often lining up with the “25 Global Biodiversity Hotspots” shown in Figure 3. For instance, Gonzalez gives Madagascar, New Zealand, and the Caucasus Mountains their own clusters, separates the Pacific Northwest from the rest of the Rockies, and divides up the Amazonian, Congo, and Indonesian rainforests in roughly canonical ways. Also, the Aleutian and Canary Islands, Newfoundland, and Hawaii are given their own clusters, and the Andes, Himalayas, and other mountain chains are not too broken up.

On the other hand, areas with low density of endangered species, such as Siberia or the Sahara Desert, seem to get partitioned in fairly meaningless ways. Also, my clusters assign the Mediterranean Basin, somewhat oddly, to 3 different centers.

But if my clustering inspires confidence by reflecting the common knowledge in some areas, then perhaps where it deviates from the usual boundaries, the results should not be immediately dismissed. They might provide questions for further research. For instance, is there really a significant difference in the species and habitats of the western half of the Mediterranean Basin versus the half that lies to the east of Sicily? What (if anything) justifies the separation of India from Indochina, Scotland from the rest of the U.K., or the area along the coast of West Africa from the Congo River Basin at a right angle to the south?

My clusters may or may not help shed light on such questions, but they do at least provide a fresh way of looking at the IUCN data. For comparison, Figures 3 through 6 provide a sample of other people’s maps of endangered species density using IUCN Red List data.

Jadon Wagstaff

Since one of the questions we have about the data is “where are areas of high biodiversity?” a density based clustering algorithm is a natural choice. Additionally a paper on the topic by () gives evidence that density based algorithms are good option for solving polygon clustering problems. Using the centroid data that Christian created, the Haversine distance metric, and suggestions from (), I implemented a modified version of dbscan in C++.

DbscanModified

epsilon ← a chosen distance

minPts ← a chosen number of centroids to find within epsilon of a centroid

p[1..n] ← set of all centroids

for all 1<=j<=n

if d[j] is unlabeled

c ← set of all centroids with distance less than epsilon around p[j]

if |c| >= minPts

label p[j] cluster alpha\_j

nbhdJump(c , alpha\_j)

else

p[j] ← outlier

return labeled set of centroids with each value of alpha corresponding to a cluster

nbhdJump( cluster c[1..m], cluster label alpha^\*)

for all 1<=i<=m

if c[i] is unlabeled or labeled outlier

label c[i] cluster alpha^\*

c' ← set of all centroids with distance less than epsilon around c[i]

if |c'| >= minPts

nbhdJump(c', alpha^\*)

Run-time for the implementation on all 74,529 centroids was about half an hour. The results are displayed in figure (). An epsilon radius of 75km seemed to produce good clusters, but more inquiry is required to determine what is an optimal value.

Another topic of conversation in () is desirable methods of measuring distance between two polygons A and B. They show that Hausdorf Distance H has desirable qualities and demonstrate its usefulness in their own implementation. The problem we face is that each of the polygons in our dataset are quite complex, so we need to determine if there is a viable implementation method for calculating these distances.

H(A,B) = max(max a ( min b d(a,b) ), max b ( min a d(a,b) ))

Here are the things I would like to accomplish going forward:

- Determine feasibility of implementing Hausdorf distance for dbscan.

- Implement the modified dbscan on the set of all terrestrial animals using Hausdorf and/or centroids.

- Query IUCN database to determine which polygons represent species on the red list.

- Use best implementation method on red list species.

Appendix:

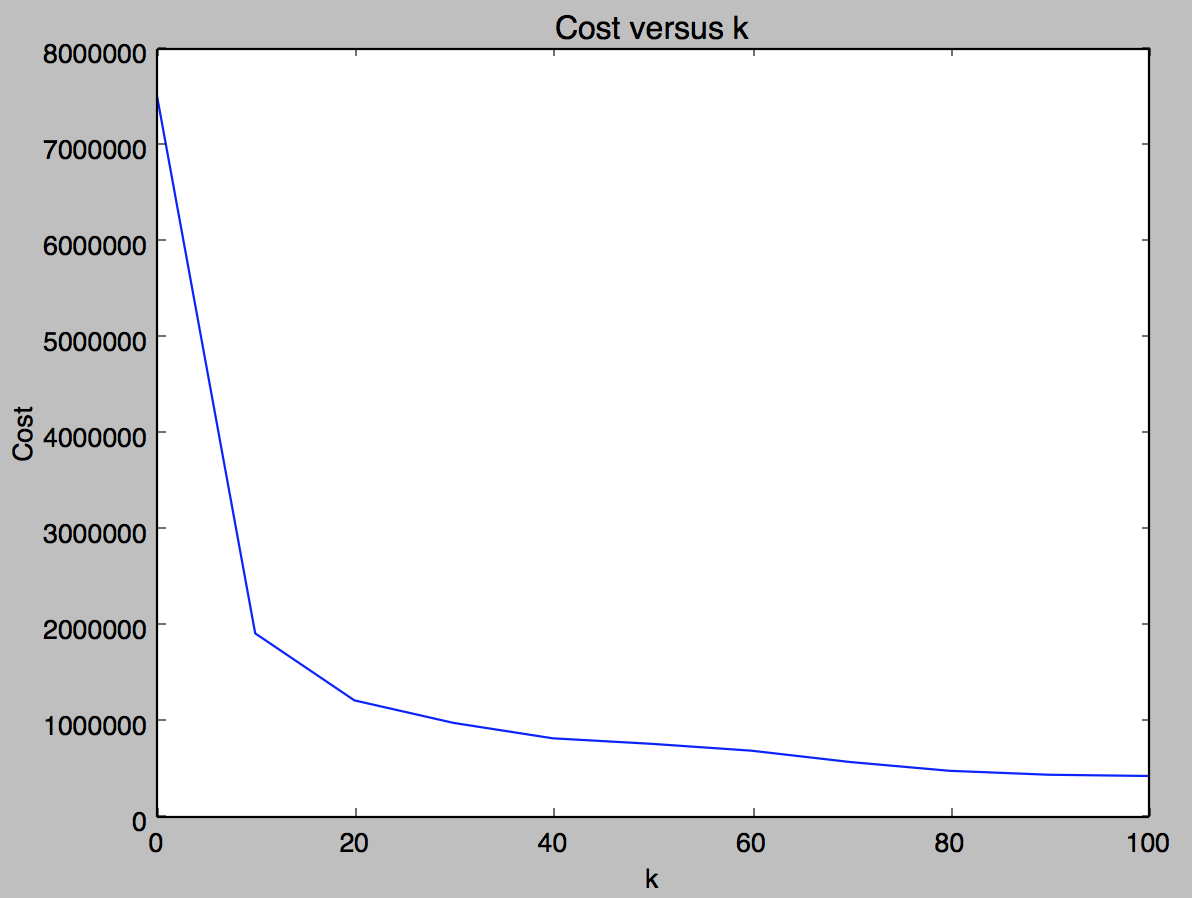


Figure 1: k-means cost in meters for Gonzalez



Figure 2: Gonzalez with k=40

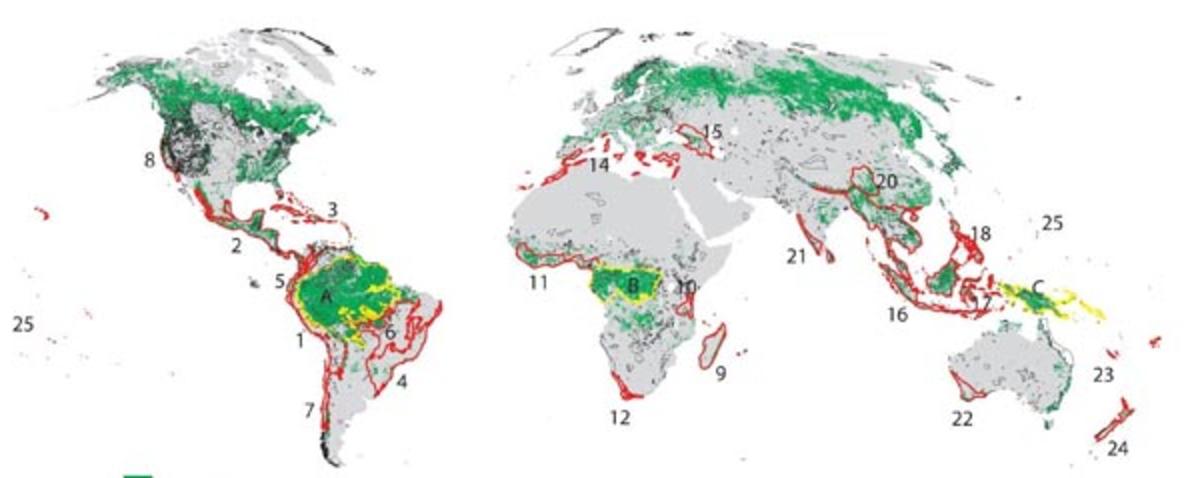


Figure 3: 25 Biodiversity “Hot Spots” from lesson plans available at <https://myweb.rollins.edu/jsiry/USAENDA.html>

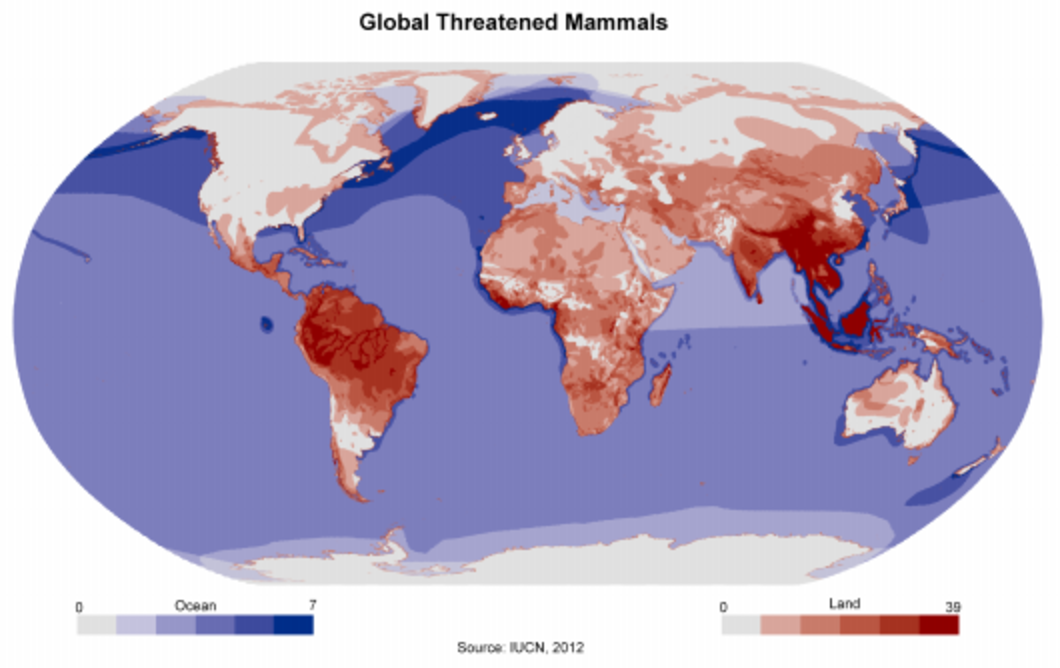


Figure 4: From Nitin Saxena's (works at National Thermal Power Corporation) maps on quora: <https://qph.ec.quoracdn.net/main-qimg-5f82bd6271b72e81c6de32e0afc8b1ed>

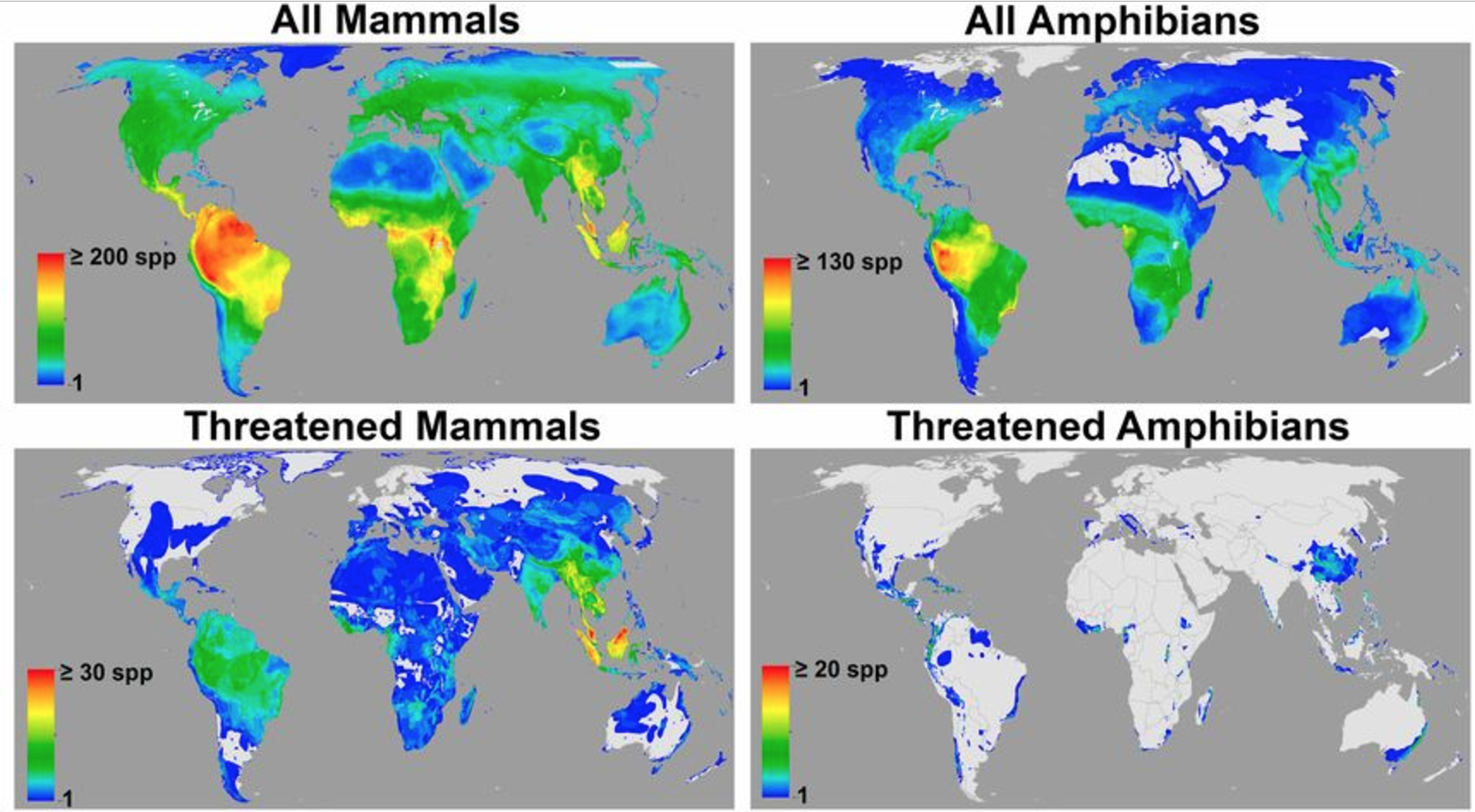


Figure 5: From Jenkins, Pimm, Joppa, "Global patterns of terrestrial vertebrate diversity and conservation", 2013. Figures accessible at <http://www.pnas.org/content/110/28/E2602.figures-only>

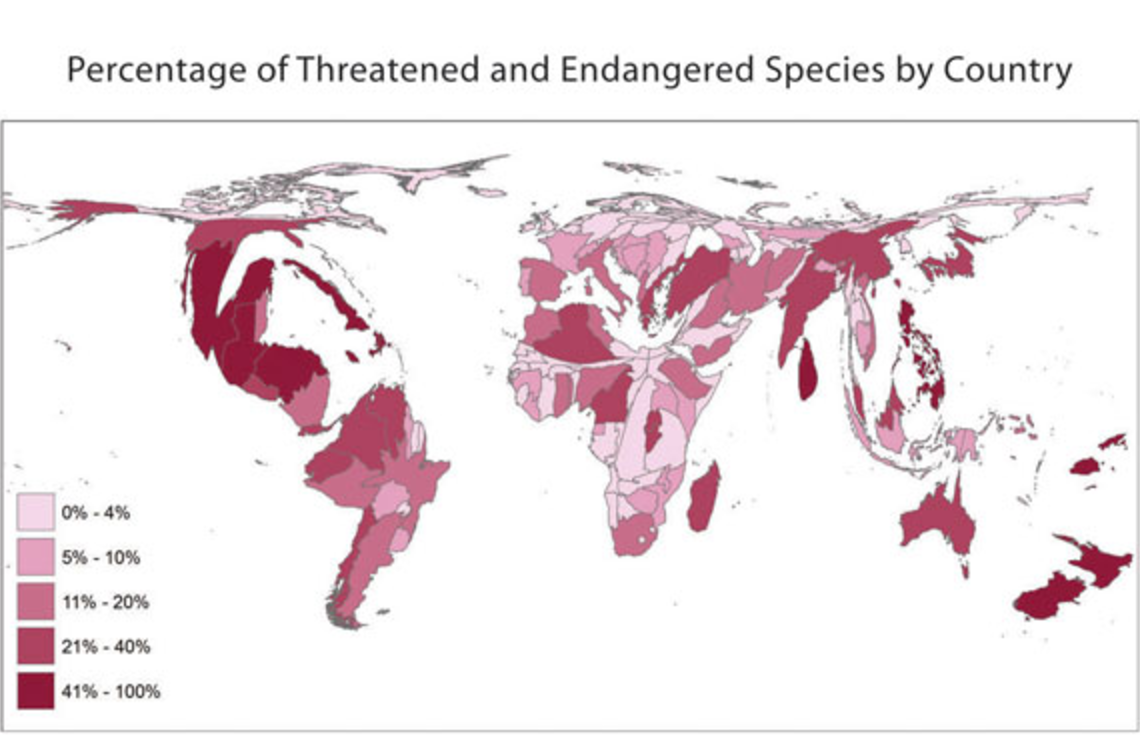


Figure 6: Amphibian data from "Visualizing AmphibiaWeb Data with Continuous Cartograms" by Koo, Vredenburg, et al. at http://www.amphibiaweb.org/amphibian/cartograms/

