Predicting Internet Access Using a Neural Network
(With a Little Help From CartoDB)

Approach

- The CartoDB project predicted the joint probability of a person in the US population having both a sight problem and vision problem at the Block Group level.

- Our project is slightly simpler - we only care about predicting the probability of a single value (internet access).

Method Overview

- The way it works is that first we create a predictive model for a target variable (internet access) based on the known input data at the PUMAS scale. We can then take our model and provide inputs at the Block Group scale and create new outputs for our new desired scale.

- Our target variable (internet access) is most likely related to other PUMs variables such as age and income, so we'll create a Neural Network to capture these correlations.

- After training the model we will want to check that it is capable of accurately predicting what we trained it to predict. To do this we will simply get the model to predict the values of a handful of PUMAS areas that we held back from the model while training. If the predicted values are close to the known ones then the model is doing a good job.

Method Step Specifics

1. Count the total number of people WITH and WITHOUT internet access for each PUMA. We'll end up with something like:

PUMA10	Internet Access	No Internet Access
100	531	836
101	294	175
102	849	395

2. Get the rest of the model features from the PUMAS dataset for each PUMA10.

3. Normalize the data.

4. Split data in training/test.

5. Train a Neural Network on the data (CartoDB uses the Keras library for this, but we can probably just use the code that Martin gave us).

6. Use our newly trained model to generate outputs based on Block Group level data.

References

- http://blog.cartodb.com/creating-segments-from-the-census/
- http://nbviewer.jupyter.org/gist/stuartlynn/8836db0a05ea48a26847
- http://keras.io/