A Deep Learning Approach to Predicting Race Using Personal Name and Location (Natural Language Processing)

Nadia Kotova
nkotova@stanford.edu
SUNet ID: nkotova

Abstract

In this project, I train a recurrent neural network to predict individuals' race using information contained in their name and location of residence. I introduce a novel data source that contains millions of race/name/zipcode triplets and covers the entire US. I train my baseline LSTM model using only personal name data. The baseline model attains overall accuracy of 0.87, which is a non-trivial improvement over the existing benchmarks in the literature. However, because personal names of Non-Hispanic Whites and Non-Hispanic Blacks follow similar naming conventions, the baseline model struggles at accurately classifying Non-Hispanic Blacks. Using location data in addition to personal name improves classification accuracy of the model for Non-Hispanic Blacks substantially, and allows the best model to achieve 0.89 accuracy on the test set.

Introduction

Personal names have become one of the most easy-to-access sources of data, available to both researchers and others. However, many of such databases do not contain other information, such as racial identifiers, that could be useful in many contexts. Being able to supplement existing databases with racial information using data on personal names and coarse locations (such as zipcodes) is of great relevance for marketing, healthcare, research on racial inequalities, and working with public databases with missing data. In fact, results of previous racial/ethnicity classification studies have been extensively used by researchers across many fields, e.g. Wu et al. (2014), Humphreys et al. (2016), Chang et al. (2010). In this project, I train a recurrent neural network to predict race using information contained in the name and zipcode location of a person. I find that a baseline model based only on individuals' last names achieves a high prediction accuracy of 0.87, but performs extremely poorly on Non-Hispanic Blacks (f1-score of 0.04). A model based on both first and last names raises the f1-score for Non-Hispanic Blacks to 0.17, while achieving the same overall accuracy of 0.87. Finally, when the model takes into account both first and last names, in addition to an individual's location, the model gives the best results: it has f1-score for Non-Hispanic Blacks of 0.47, and achieves a 0.89 accuracy overall. In previous work, Sood and Laohaprapanon (2018) train a recurrent neural network to predict the race of each personal name using Florida Voter Registration data and Wikipedia data, and acheieve an accuracy of 0.84 in their best model. As such, my results are a substantial improvement over the benchmarks in the literature.

Literature review

The project contributes to the literature on using machine learning frameworks to predict race, ethnicity, and nationality. Ambekar et al. (2009) combine decision tree and Hidden Markov Model to perform classification with 13 ethnic categories. Treeratpituk and Giles (2012) train a multinomial logistic regression and utilize both alphabetical and phonetic sequences in names to achieve more accurate performance. Lee et al. (2017) train a recurrent neural network-based model to predict nationalities of each name using Olympic record data. Sood and Laohaprapanon (2018) also train a recurrent neural network to predict the race of each personal name using Florida Voter Registration data and Wikipedia data. Imai and Khanna (2016) and Wong et al. (2020) find that using input features based on both name and location further improves the performance.

The contribution of this project is two-fold. First, I introduce a novel data source that has several advantages over datasets used in the literature. In particular, previously used databases either have only last names (Census), have relatively few observations (Wikipedia data), or may not be representative of the entire US (Florida Voter Registration Data). Second, I leverage zip code information contained in my dataset to improve classification accuracy relative to the literature benchmarks.

Data

The data for this project comes from two main sources. The first is the Home Mortgage Disclosure Act (HMDA) public dataset. The HMDA contains records for millions of US mortgage applications which include applicants' race. The second data source is the data on the universe of US housing deeds provided by CoreLogic. Among other variables, CoreLogic records buyer and seller first and last names, as well as the house location. By merging two datasets and matching mortgage applications with the corresponding housing transactions, I extract millions of name/race/zipcode triplets. My final dataset contains over 16M observations on Asian, Hispanic, Non-Hispanic Black, and Non-Hispanic White individuals (see Table 1).

Race	Observations	Frequency
Asian	1,194,234	7.3%
Hispanic	1,654,942	10.2%
Non-Hispanic Black	883,825	5.4%
Non-Hispanic White	12,541,523	77.1%

Table 1: Number of observations and frequency by race

Baseline model

In my baseline model, I only use data on personal first and last names. I randomly sample test and development sets from the original dataset. Both test set and development set have 50K observations. For my training set, I randomly sample 2M observations. To learn the relationship between personal name and race, I am using an LSTM model from Sood and Laohaprapanon (2018) as my baseline. First, I concatenate the capitalized last and first names, and then split the strings into two character n-grams. I then follow Sood and Laohaprapanon (2018) and remove n-grams that occur less than 3 times in the data and n-grams that occur in over 30% of the strings in the data. My vocabulary consists of 1259 remaining n-grams. I then implement one-hot encoding to represent each string as a sequence of integers. Next, I make sure that all sequences have the same size. I pad sequences with zeros if necessary such that each sequence has a length of 25.

Before estimating the model, I embed each of the two character n-grams into a real-valued vector of length 32. The baseline model architecture then consists of the following layers: an LSTM with 128 hidden units, a dropout layer with 0.8 keep probability, a final dense layer with a softmax activation function. Figure 1 further describes the model.

Next, I use Keras to estimate the model. Because it is a classification problem with multiple classes, I use categorical cross-entropy as the loss function. I fit the model using the ADAM optimization algorithm with a batch size of 512. I iterate for 7 epochs. Table 2 presents the results on the development and test sets. For reference, Table 3 presents Table 4 from Sood and Laohaprapanon (2018), which summarizes the performance of their LSTM model.

Race	Precision	Recall	F1-score	Support
Asian	0.82	0.70	0.76	3,637
Hispanic	0.79	0.72	0.75	5,207
Non-Hispanic Black	0.62	0.09	0.16	2,714
Non-Hispanic White	0.89	0.96	0.92	38,442
Average	0.78	0.62	0.65	50,000
Weighted average	0.86	0.87	0.85	50,000
Accuracy			0.87	50,000

Table 2: Performance of the baseline model trained on last and first names on the dev set

The model achieves an accuracy of 0.87 overall. Relative to the Sood and Laohaprapanon (2018) benchmark, my baseline model's performance is better for Asians and Non-Hispanic Whites, similar for Hispanics, and much worse for

Race	Precision	Recall	F1-score	Support
Asian	0.77	0.49	0.6	4,527
Hispanic	0.76	0.73	0.75	18,440
Non-Hispanic Black	0.73	0.43	0.55	28,586
Non-Hispanic White	0.86	0.84	0.83	146,009
Weighted average	0.83	0.84	0.83	197,562
Accuracy			0.84	197,562

Table 3: Performance of the LSTM model by Sood and Laohaprapanon (2018)

Label\Prediction	Asian	Hispanic	Non-Hispanic Black	Non-Hispanic White
Asian	2549	157	12	919
Hispanic	41	3754	18	1394
Non-Hispanic Black	54	56	244	2360
Non-Hispanic White	464	800	120	37058

Table 4: Confusion matrix for the development set

Non-Hispanic Blacks. As evident from Table 4, most Non-Hispanic Blacks are classified as Non-Hispanic Whites by the baseline model. Nevertheless, my baseline model achieves a higher overall accuracy than those reported in Sood and Laohaprapanon (2018). Since the architecture of my baseline model is identical to Sood and Laohaprapanon (2018), the differences in the results are most likely due to differences in the underlying data sources. Sood and Laohaprapanon (2018) use Florida Voter Registration Data. Since naming conventions in Florida might not be representative of the entire US, the much better results in Sood and Laohaprapanon (2018) for Non-Hispanic Blacks might not be generalizable beyond Florida. Another potential explanation is that the training sample in Sood and Laohaprapanon (2018) contains relatively more Non-Hispanic Black individuals. I deal with the class imbalance problem in my dataset in the next Section.

Class imbalance

The main challenge going further is attaining higher prediction accuracy for African Americans, whose last names are similar to non-Hispanic whites (Bertrand and Mullainathan (2004)). Notice that Non-Hispanic Whites represent 77.1% of the entire dataset. It is possible that feeding more Asian, Hispanic, and Non-Hispanic Black examples into the network could improve its performance. In this section, I address class imbalance in my dataset by creating a perfectly balanced training sample. For each race class, I sample 500K observations into my training set. Therefore, I keep the total number of training samples in my dataset fixed. For consistency, I evaluate the performance of all models on the same (unbalanced) development set. Table 5 reports the performance of the baseline model on the development set. The overall performance of the baseline model trained on the balanced training set is considerably worse than of the baseline model trained on the original training set. Even though the classification accuracy is much higher for Non-Hispanic Blacks, it is much lower for Non-Hispanic Whites. Qualitatively, these results suggest that the model performs poorly at classifying Non-Hispanic Blacks because their names are hard to distinguish from Non-Hispanic Whites, and not because of the class imbalance in the original dataset.

Alternative RNN architectures

In this section, I examine a different network architecture. As before, I embed each of the two character n-grams into a real-valued vector of length 32. The alternative model architecture then consists of the following layers: a 1D CNN with 32 filters of size 3x3, same padding, and ReLU activation, a 1D Max Pooling layer with a 2x2 filter, an LSTM with 100 units, and a final dense layer with a softmax activation function. I fit the model using the ADAM optimization algorithm and a batch size of 512. I iterate for 7 epochs. Figure 2 further describes the model's architecture. Table 6 reports the

Race	Precision	Recall	F1-score	Support
Asian	0.62	0.79	0.70	3,637
Hispanic	0.60	0.82	0.70	5,207
Non-Hispanic Black	0.16	0.65	0.26	2,714
Non-Hispanic White	0.94	0.66	0.78	38,442
Average	0.58	0.73	0.61	50,000
Weighted average	0.84	0.69	0.73	50,000
Accuracy			0.69	50,000

Table 5: Performance of the baseline model on the dev set (fully balanced training set).

performance of the model on the development set. Notice that the alternative model achieves slightly worse results than the baseline model, but takes much less time to converge, and, therefore, could prove useful in future work.

Race	Precision	Recall	F1-score	Support
Asian	0.79	0.71	0.75	3,637
Hispanic	0.79	0.71	0.75	5,207
Non-Hispanic Black	0.61	0.08	0.14	2,714
Non-Hispanic White	0.89	0.96	0.92	38,442
Average	0.77	0.61	0.64	50,000
Weighted average	0.85	0.87	0.85	50,000
Accuracy			0.87	50,000

Table 6: Performance of the CONV+MaxPool+LSTM model on the development set

Using location data: passing zipcode as a string

In this section, I propose to exploit location data to achieve better network performance. Glaeser and Vigdor (2001) document that the majority of black Americans continue to live in predominantly black locations. The distributions of Asian and Hispanic populations is uneven across space as well, therefore, geolocational data can be informative at predicting race. I first concatenate the capitalized last and first names with zipcode strings. I then apply the same data pre-processing procedure as for my baseline model, except in the end I pad each sequence to have a length of 31 instead of 25 to take into account that my data strings are now longer because of the additional zipcode information. Table 7 reports the results on the development set. Using zipcode location improves the results for Non-Hispanic Blacks: F1-score increases from 0.16 (baseline) to 0.26 (with zipcode).

Race	Precision	Recall	F1-score	Support
Asian	0.82	0.69	0.75	3,637
Hispanic	0.79	0.72	0.75	5,207
Non-Hispanic Black	0.58	0.16	0.26	2,714
Non-Hispanic White	0.89	0.96	0.92	38,442
Average	0.77	0.63	0.67	50,000
Weighted average	0.86	0.87	0.86	50,000
Accuracy			0.87	50,000

Table 7: Performance of the baseline model trained on personal name and zipcode string on the dev set.

Using location data: using numerical data on racial composition of zipcodes

I further investigate whether I can improve the model's performance by directly supplying numerical data on the racial composition of each zipcode. I modify the network architecture to accomodate the numerical features. I first pass personal name strings through an embedding layer and LSTM layer with 128 hidden units, then I concatenate the output with numerical input features, then add a dense layer with 32 hidden units and a final softmax layer to the network. Figure 3 describes the network architecture. Table 8 reports the performance of the model on the development set. The results show that using data on zipcode racial composition drastically improves classification accuracy for Non-Hispanic Blacks, as indicated by the increase in the F1-score from 0.16 to 0.47.

Race	Precision	Recall	F1-score	Support
Asian	0.83	0.74	0.78	3,637
Hispanic	0.78	0.75	0.77	5,207
Non-Hispanic Black	0.65	0.37	0.47	2,714
Non-Hispanic White	0.91	0.95	0.93	38,442
Average	0.79	0.70	0.74	50,000
Weighted average	0.88	0.89	0.88	50,000
Accuracy			0.89	50,000

Table 8: Performance of the baseline model trained on personal name and zipcode racial composition on the dev set.

Using only last name data

In this subsection, I assess the importance of using data on first names for the prediction task. I train the baseline model on the data that contains *only* personal last names. Relative to the baseline model trained on both last name and first name, the last name model performs very well. In fact, it achieves the same accuracy. However, the performance of the last name model on Non-Hispanic Blacks is much worse. These results suggest that Non-Hispanic Black first names convey important information that helps the baseline model to distinguish them from Non-Hispanic Whites. See the results presented in Table 9.

Race	Precision	Recall	F1-score	Support
Asian	0.84	0.66	0.74	3,637
Hispanic	0.77	0.76	0.77	5,207
Non-Hispanic Black	0.68	0.02	0.04	2,714
Non-Hispanic White	0.88	0.96	0.92	38,442
Average	0.79	0.60	0.62	50,000
Weighted average	0.86	0.87	0.84	50,000
Accuracy			0.87	50,000

Table 9: Performance of the baseline model trained on only last name on the dev set.

Discussion

The performance on the test set of the baseline model trained on last name only, last name and first name, and last name, first name and zipcode racial composition is similar to their performance on the respective dev sets. I report the test set results in the Appendix, Tables 10, 11, 12. The results show that having data on personal last names is already enough to get fairly accurate predictions for Asians, Hispanics and Non-Hispanic Whites. However, data on first names and locations is crucial for achieving acceptable results for Non-Hispanic Blacks.

References

Ambekar, Anurag, Charles Ward, Jahangir Mohammed, Swapna Male, and Steven Skiena, "Name-ethnicity classification from open sources," in "Proceedings of the 15th ACM SIGKDD international conference on Knowledge Discovery and Data Mining" 2009, pp. 49–58.

Bertrand, Marianne and Sendhil Mullainathan, "Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination," *American economic review*, 2004, *94* (4), 991–1013.

Chang, Jonathan, Itamar Rosenn, Lars Backstrom, and Cameron Marlow, "epluribus: Ethnicity on social networks," in "Proceedings of the International AAAI Conference on Web and Social Media," Vol. 4 2010.

Humphreys, Brad R, Adam Nowak, and Yang Zhou, "Cultural superstitions and residential real estate prices: Transaction-level evidence from the us housing market," *Available at SSRN 2890655*, 2016.

Imai, Kosuke and Kabir Khanna, "Improving ecological inference by predicting individual ethnicity from voter registration records," *Political Analysis*, 2016, pp. 263–272.

Lee, Jinhyuk, Hyunjae Kim, Miyoung Ko, Donghee Choi, Jaehoon Choi, and Jaewoo Kang, "Name Nationality Classification with Recurrent Neural Networks.," in "IJCAI" 2017, pp. 2081–2087.

Sood, Gaurav and Suriyan Laohaprapanon, "Predicting race and ethnicity from the sequence of characters in a name," *arXiv preprint arXiv:1805.02109*, 2018.

Treeratpituk, Pucktada and C Lee Giles, "Name-ethnicity classification and ethnicity-sensitive name matching," in "Proceedings of the AAAI Conference on Artificial Intelligence," Vol. 26 2012.

Wong, Kai On, Osmar R Zaïane, Faith G Davis, and Yutaka Yasui, "A machine learning approach to predict ethnicity using personal name and census location in Canada," *Plos one*, 2020, *15* (11), e0241239.

Wu, Zhaohui, Dayu Yuan, Pucktada Treeratpituk, and C Lee Giles, "Science and ethnicity: How ethnicities shape the evolution of computer science research community," *arXiv preprint arXiv:1411.1129*, 2014.

Appendix

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 25, 32)	40544
lstm_1 (LSTM)	(None, 128)	82432
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 4)	516

Total params: 123,492
Trainable params: 123,492
Non-trainable params: 0

Figure 1: Baseline model architecture

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 25, 32)            40288
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 25, 32)            3104
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 12, 32)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               53200
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 404
=================================================================
Total params: 96,996
Trainable params: 96,996
Non-trainable params: 0
```

Figure 2: Alternative model architecture: Conv+MaxPooling+LSTM

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 25)           0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 25, 32)       40288       input_1[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 128)          82432       embedding_1[0][0]
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 3)            0
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 131)          0           lstm_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 32)           4224        concatenate_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 4)            132         dense_1[0][0]
==================================================================================================
Total params: 127,076
Trainable params: 127,076
```

Figure 3: Architecture for LSTM model trained on personal name and zipcode composition

Race	Precision	Recall	F1-score	Support
Asian	0.83	0.70	0.76	3,769
Hispanic	0.78	0.72	0.75	5,066
Non-Hispanic Black	0.65	0.10	0.17	2,656
Non-Hispanic White	0.89	0.96	0.92	38,509
Average	0.79	0.62	0.65	50,000
Weighted average	0.86	0.87	0.85	50,000
Accuracy			0.87	50,000

Table 10: Performance of the baseline model trained on last and first names on the test set

Race	Precision	Recall	F1-score	Support
Asian	0.84	0.75	0.79	3,769
Hispanic	0.78	0.74	0.76	5,006
Non-Hispanic Black	0.66	0.36	0.47	2,656
Non-Hispanic White	0.91	0.96	0.93	38,509
Average	0.80	0.70	0.74	50,000
Weighted average	0.88	0.89	0.88	50,000
Accuracy			0.89	50,000

Table 11: Performance of the baseline model trained on personal name and zipcode racial composition on the test set.

Race	Precision	Recall	F1-score	Support
Asian	0.84	0.67	0.74	3,769
Hispanic	0.76	0.75	0.75	5,006
Non-Hispanic Black	0.69	0.02	0.04	2,656
Non-Hispanic White	0.88	0.96	0.92	38,509
Average	0.79	0.60	0.62	50,000
Weighted average	0.86	0.87	0.84	50,000
Accuracy			0.87	50,000

Table 12: Performance of the baseline model trained on only last name on the test set.