ORS: The Future of Autocoding

Office of Compensation and Working Conditions

Nicole Nestoriak — Senior Research Economist

CIVIC DIGITAL FELLOWSHIP

U.S. BUREAU OF LABOR STATISTICS

Somya Jain
UC Berkeley
Cognitive Science and
Data Science

What is machine learning?

- You’ve heard the phrase so many times... but what does it mean?
- Machine learning: programming computers to optimize a performance criterion using example data or past experience
- Role of statistics: inference from a sample
- Role of computer science: efficient algorithms that →
    - Solve the optimization problem
    - Representing and evaluating the model from inference

CIVIC DIGITAL FELLOWSHIP

Objective

Our goal is to improve the current autocoder built from the ORS data by supplementing the model with data provided by Burning Glass Technologies.

The ORS autocoder

- Purpose: to develop a proof-of-concept SOC autocoder program that can be used in the ORS program to improve the accuracy of occupational classification decisions
- SOII autocoder's accuracy: around 78% (ORS prototypes: peaked at 56.7%)
- Limits
    - ORS is a relatively new program, fewer rotation groups
    - Each rotation group: available sample sizes much smaller than data used by SOII and OES autocoders
    - SOII and OES autocoders have objective gold standards for judging machine vs. human accuracy – ORS currently does not

CIVIC DIGITAL FELLOWSHIP

The ORS autocoder

- Modeled somewhat after its predecessors (i.e. SOII's SOC autocoder)
- Prototype 1
  - Mimics features used by SOII's 2014 autocoder
  - Features: job title, NAICS, ownership, state FIPS, company name, establishment
- Prototype 2
  - Use just ORS and ORS-specific features
  - Features: same as #1, but adding in ORS-specific factors (i.e. task lists)
- Prototype 3
  - Supplementing ORS data with other data sources
  - Using #1 as a starting point, then including in data sources from programs such as SOII and OES

Prototypes: results

- Prototype 1
  - Least promising results
  - Accuracy levels: lower 50's
- Prototype 2
  - Showed the best results of correct prediction for common SOCs, but not rare SOCs
  - Factors to consider: change in scope, sample design, data structure
- Prototype 3
  - Shows results that perform close to those of Prototype 2
  - Most practical form of implementation
  - Shows best results of correct prediction on average for all SOC codes, including the rare ones

Approaches

- Approach 1
  - Train model on combined datasets with shared features
- Approach 2
  - Train model on combined datasets with all features
  - Includes those that are not shared
- Approach 3
  - Model 1:
    - Train solely on Burning Glass dataset
  - Model 2:
    - Train only on the ORS data, but use outputs from Model 1 as inputs to Model 2

What we have accomplished

- Have built a model to fit and train our datasets on
- Feature extraction
    - Using CountVectorizer to convert our raw input data into vectors
    - Then, transformed the vectors to stack them into a matrix
- Models: for fitting and training the data
    - Logistic regression
    - SGD classifier
        - Stochastic gradient descent: select a random sample from the training data and iterates it by trying to find minimums and maximums in the data
        - Especially good for large datasets
    - Random forest

What we have accomplished

- Minimum document frequency
  - By default: CountVectorizer includes any feature (i.e. word) that occurs in our training data as a feature in our feature vectors
  - Can use MDF to specify that features should only be included in the feature vector if they occur in at least ”X” different documents
  - Good if you want to decrease the number of features you plan on using
- n-grams
  - Another default: only include individual words as features
  - What about phrases or other short sequences of words?
    - 2-word sequences = bigrams
    - 3-word sequences = trigrams
  - Can use n-grams to tell our vectorizer to create features for all individual words and word pairs that occur in at least “X” examples in our training set
  - Good if you want to increase the number of features you plan on using

What we have learned so far

- Logistic regression → good!
  - Limits: logistic regression may take too long on large datasets, even though adding more data to the model may increase accuracy rates
  - Would recommend using logistic regression on datasets that operate on a smaller scale
- Stochastic gradient descent → good!
- Model's performance on the Burning Glass dataset on SOC codes:
  - Accuracy on training set: 89.66%
  - Accuracy on validation set: 89.52%
- Model's performance on the ORS dataset without supplementing it with Burning Glass data on SOC codes
  - Accuracy on training set: 86.25%
  - Accuracy on validation set: 32.98%

CIVIC DIGITAL FELLOWSHIP

What we have learned so far

- Model's performance on a combined ORS and Burning Glass dataset with all features using NAICS codes:
  - Accuracy on training set: 99.78%
  - Accuracy on validation set: 99.71%
- Model's performance on a combined ORS and Burning Glass dataset with all features using job titles:
  - Accuracy on training set: 99.69%
  - Accuracy on validation set: 99.65%

CIVIC DIGITAL FELLOWSHIP

Next steps for the future

- The model produces the highest accuracy rates for NAICS codes and job titles, with lower rates for SOC codes. Why?
- ORS is a younger program compared to, for example, SOII and OES
  - A lot of our limitations stem from how new the program really is
- Possible application of word embedding techniques, such as word2vec
- While our accuracy levels have increased considerably, the next goal would be to bump the SOC rates up to 90% or higher
- Developing another gold standard dataset for ORS?
- Looking into other data sources?
  - The public sector vs. private sector question

CIVIC DIGITAL FELLOWSHIP

THANK YOU!

- Nicole Nestoriak, Bureau of Labor Statistics
- Alex Measure, Bureau of Labor Statistics
- Brandon Kopp, Bureau of Labor Statistics
- Jennifer Edgar, Bureau of Labor Statistics
- Kristen Monaco, Bureau of Labor Statistics
- Rachel Dodell, Coding It Forward
- Chris Kuang, Coding It Forward
- Hillary Mclauchlin, Coding It Forward