Wu's Replication Package Read Me

Jesse Schmolze

High Level Overview

Run the replication package by doing the following:

1. Run Lasso-logit-full-sample.py. It takes around 15 minutes.
2. Run lasso-logit-pronoun-sample.py. It takes around 15 minutes.
3. Run lasso-linear-pronoun-sample.py. It takes around 15 minutes.
4. Retrieve tables and data using standard methods in tables-figures.R.
5. Run Naive_Bayes.py. It takes 10 seconds. The table outputed in the terminal is ready for use.

1 Description of Datasets

The following datasets are derived from a four-year sample of the Economics Job Market Rumors (EJMR) forum.

- **gendered_posts.csv**: A dataset of Female/Male posts identified from the four-year EJMR sample.
- **vocab10K.csv**: A list of the 10,000 most frequent words from 2.2 million posts (Oct 2013–Oct 2017) and their marginal probabilities from Lasso models[cite: 8, 9].
- **X_word_count.npz**: A sparse matrix recording occurrences of the top 10,000 words in each post, used for logistic/linear Lasso models[cite: 11, 12, 13].
- **keys_to_X.csv**: Unique identifiers (`title_id` and `post_id`) for merging with the word count matrix[cite: 14, 15, 16].
- **trend_stats.csv**: Monthly summary statistics for trend analysis[cite: 17, 19].

2 Original Lasso Programs

The original analysis utilizes the following Python and R scripts:

1. **lasso-logit-full-sample.py**: Logistic Lasso on the full gender sample.
2. **lasso-logit-pronoun-sample.py**: Logistic Lasso on the pronoun-based sample.
3. **lasso-linear-pronoun-sample.py**: Linear Lasso on the pronoun sample.
4. **tables-figures.R**: Constructing final tables and figures.

3 Naive Bayes Extension

In addition to the original Lasso-regularized models, a Naive Bayes classifier was implemented to identify distinctive gendered language using log-probability ratios.

3.1 Optimization and Data Filtering

Prior to running the algorithm, restrictions were applied to optimize the dataset for the Naive Bayes model. The following steps were taken before running the algorithm:

- **Log-Odds Calculation**: The model evaluates the ratio of the log probability of seeing male words in male posts minus the log probability of seeing female words in female posts. Specifically, the log-odds ratio is calculated by subtracting male log-probabilities from female ones; a high positive score identifies female-leaning terms, while a high negative score identifies male-leaning terms.

- **Noise Reduction**: A Regex filter was applied to retain only standard English characters (`a-z`).

- **Length Constraint**: A minimum length of 3 letters was enforced to eliminate short, non-distinctive strings and random noise.

3.2 Implementation Code

The Python Code can be found in Naive_Bayes.py

4 Codebook Summaries

4.1 Gendered Posts (gendered_posts.csv)

Variable	**Description**
title_id	Uniquely identifies a thread [cite: 26]
post_id	Uniquely identifies a post in each thread [cite: 26]
female	1 if Female, 0 if Male (Final classification) [cite: 26]
ypred	Predicted probability of a post discussing a female [cite: 26]

4.2 Vocabulary (vocab10K.csv)

Variable	**Description**
word	Most frequent 10,000 words in lower case
coef	Estimated coefficient in the Lasso-logistic model
ME	Estimated average marginal effect