MSDS 411: Unsupervised Learning

Assignment 1

July 8, 2025

Feature Engineering for Political Research

Yingyu Mao

Abstract

This study explored various unsupervised learning techniques for feature engineering on a political survey dataset from the Pew Research Center, which is characterized by its high dimensionality and binary data type. I addressed the challenges of using standard dimension reduction methods on Bernoulli-distributed binary data and compared standard Principal Component Analysis (PCA) with logistic PCA, demonstrating that logistic PCA is more effective at representing the data structure for this dataset, despite similar explained variances by their initial principal components. I also investigated hierarchical clustering of features, revealing that while this method resulted in low-quality clusters, it uncovered interesting patterns in neutral responses, potentially indicating underlying determinants. Finally, I applied Restricted Boltzmann Machine (RBM) for learning nonlinear relationships among binary features, highlighting its ability to derive latent variables and maintain interpretability. The engineered features from these methods can be used for subsequent supervised learning tasks, such as predicting voting outcomes.

Keywords: Feature Engineering, Dimension Reduction, Political Surveys, Binary Data Analysis, Unsupervised Learning

1.0 Introduction

Political research is a systematic study of political phenomena. Its goal is to explain and predict political behavior. One important research tool for data generation is political surveys, which are used to measure public opinion, voter behavior, and attitudes toward policies, political candidates, and government performance. Political survey data presents a unique set of challenges including sample bias and representativeness caused by declining response rate and sample frame bias, and high measurement error due to social desirability bias or recall bias, and more recently political polarization and distrust. Besides all these issues with survey data itself, the data being high dimensional also poses challenges with prediction and interpretation. This is where dimension reduction techniques become invaluable.

Dimension reduction is a type of unsupervised learning method that transforms high-dimensional data into a lower-dimensional space while retaining as much of the original pairwise relationship as possible. Dimension reduction helps to mitigate “curse of dimensionality” which arises with data sparsity in high dimensional space. Factor and component analyses reveal the direction of variance and identify complex multidimensional latent variables, which are not directly observable but are determined by a set of observable variables. These engineered features can be used for downstream training of supervised learning models to predict voting outcomes, for instance. These analyses can also reveal noisy or irrelevant variables that can be eliminated from downstream model training.

In this article, I will discuss several dimension reduction techniques and their usage in feature engineering for a political survey dataset published by Pew Research Center.

2.0 Methods and Literature Review

2.1 Data cleaning and encoding

The survey data generation process is detailed in a methodology report published by Pew Research Center along with the data (Schalk, et al. 2019). The survey questions and their answers were selected as political opinion variables for further analysis. All features (columns) with missing values were excluded from the analysis. The remaining features (all categorical data) were one-hot encoded. The final number of features is 128.

2.2 Dimension reduction algorithms for feature engineering

Binary data (Bernoulli distributed) could behavior very differently from continuous data when running certain algorithms. We need to identify algorithms that are suitable for binary data.

2.2.1 Principal component analysis to generalized PCA

PCA is a linear dimension reduction technique performed on the correlation matrix or covariance matrix of continuous data. Karl Pearson's formulation of PCA (Pearson 1901) aimed to find a lower-rank subspace that best represents the data points by minimizing the squared error (Euclidean distance) between the original data points and their projections onto that subspace. This is implicitly tied to the assumption of Gaussian data distribution, where the squared error is

directly related to the Gaussian negative log-likelihood. For any $n \times d$ data matrix $\boldsymbol{X}$, when each column vector $\boldsymbol{x_i}$ is centered, the goal of PCA is to minimize the negative log-likelihood $\sum_{i=1}^{n} ||\boldsymbol{x_i} - \boldsymbol{\theta_i}||^2$, where $\boldsymbol{\theta_i}$ are the projections of $\boldsymbol{x_i}$ in a k-dimensional subspace. The negative log-likelihood is Gaussian deviation. When the data is Gaussian distributed, the natural parameter of the saturated model of $\boldsymbol{X}$ is $\widetilde{\boldsymbol{\theta}}_{\boldsymbol{i}} = \boldsymbol{x_i}$. Standard PCA is in essence finding the lower-dimensional projections of the natural parameter of the saturated model.
Binary data follows Bernoulli distribution, where the observed data x_{ij} is generated from Bernoulli (p_{ij}). In the saturated model, $p_{ij} = x_{ij}$, followed by $\tilde{\theta}_{ij} = logit(p_{ij})$. To find the low-dimensional projection $\hat{\theta}_{ij} = \tilde{\theta}_{ij} UU^T$ is to minimize the corresponding Bernoulli deviance $D(X|\hat{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d} -2x_{ij}\hat{\theta}_{ij} + 2log(1 + exp(\hat{\theta}_{ij}))$, a method described as logistic PCA (Landgraf and Lee 2015). I will apply both the standard and the logistic PCA to our binary dataset and compare the results.

2.2.2 Hierarchical clustering of features

Standard PCA also assumes linear correlation between features. This might not be the case for this dataset. Agglomerative clustering (Izenman 2008) is an unsupervised learning technique that groups similar features from bottom-up iteratively and generates a tree-like mapping of the data structure. Because the agglomeration of the features is dependent upon dissimilarity or similarity metric (features of the least dissimilarity are clustered together), the metric should also be suitable for the binary dataset. Both Gower's distance (Gower 1971) and Jaccard index (Jaccard 1901) can be used to generate the distance matrix.

2.2.3 Restricted Boltzmann machine

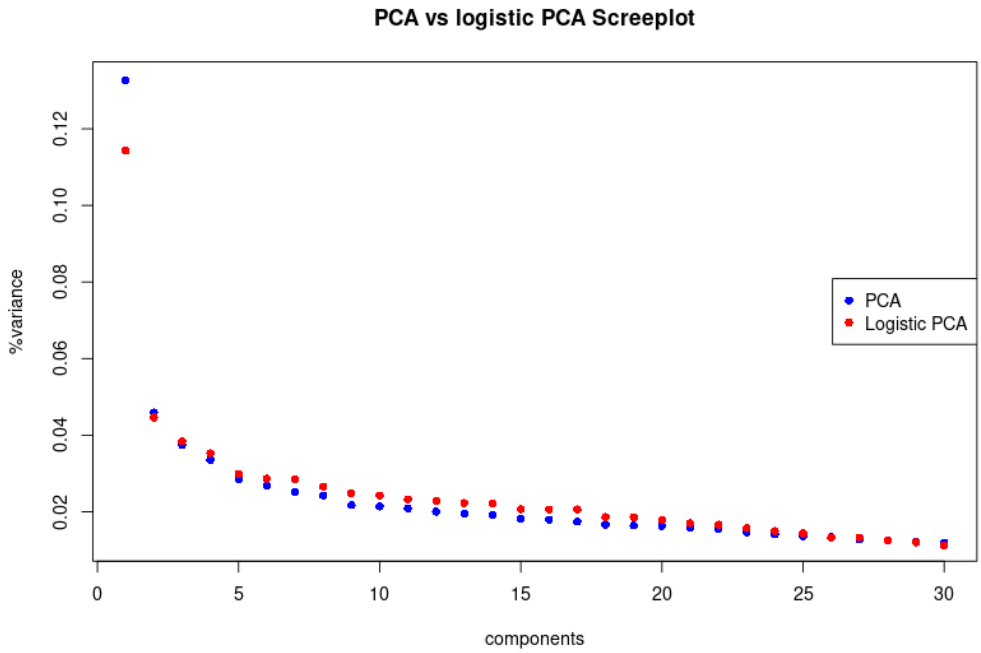
Another way to learn nonlinear relationships among features of binary data type is through generative stochastic neural network (Hinton 2002). Restricted Boltzmann machine is a variant of Boltzmann machines with the restriction that its neurons must form a weighted bipartite graph. The two groups of neurons are the visible and hidden layers, respectively. The connection strength between the hidden and visible units is represented by weights, w_{ij}. The visible units receive input data and represent measurable variables. The hidden units are fully connected only with the visible units and are therefore independent of each other. This makes them independent components that are transformations of the visible (measurable) units. After adding a nonlinear activation function (logistic sigmoid function for binary data) for the hidden units, RBM can learn the nonlinear relationships among the visible units by finding the weights that maximize the probability of the input data distribution: $\underset{W}{\operatorname{argmax}} \prod_{v \in V} P(v)$.

3.0 Model Training and Results

3.1 Comparison between standard PCA and logistic PCA

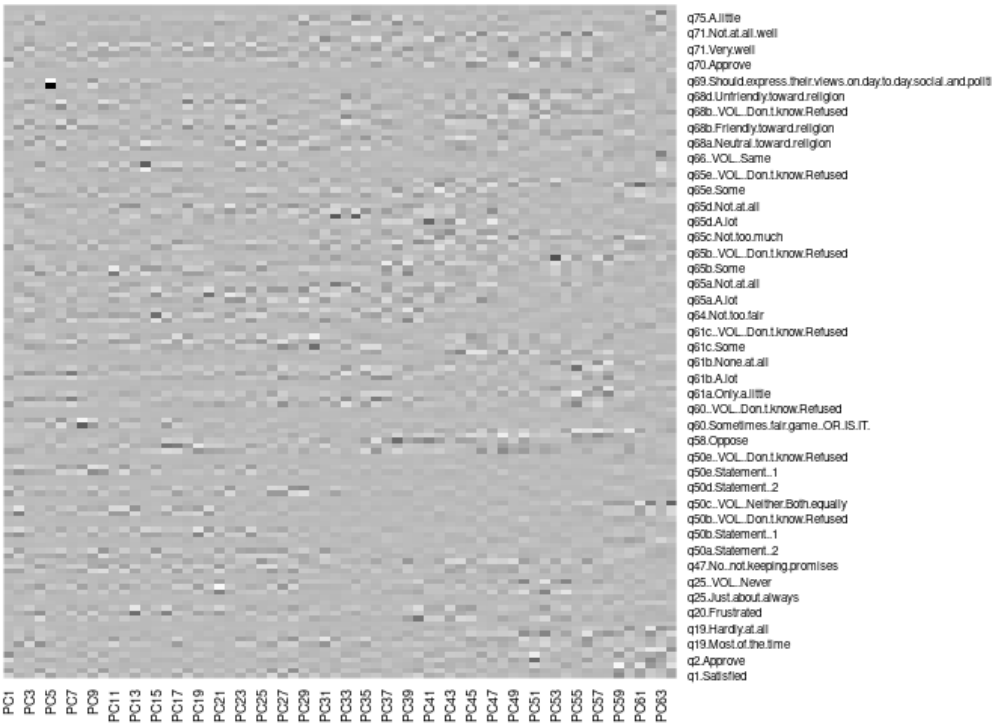
PCA can be performed with basic R and logistic PCA can be performed with R package `logisticPCA`. I ran both analyses in R.

The natural parameters for logistic PCA are $\tilde{\theta}_{ij} = \begin{cases} -\infty, & x_{ij} = 0 \\ \infty, & x_{ij} = 1 \end{cases}$, which poses computational difficulty. Therefore, it is approximated by a large number m. The value of m is a hyperparameter that can be tuned through cross validation. After testing a parameter grid of different components k and m, m is determined as 4 (data not shown).
The variances explained by each component learned through the two algorithms agree mostly with each other as shown in the scree plots below.

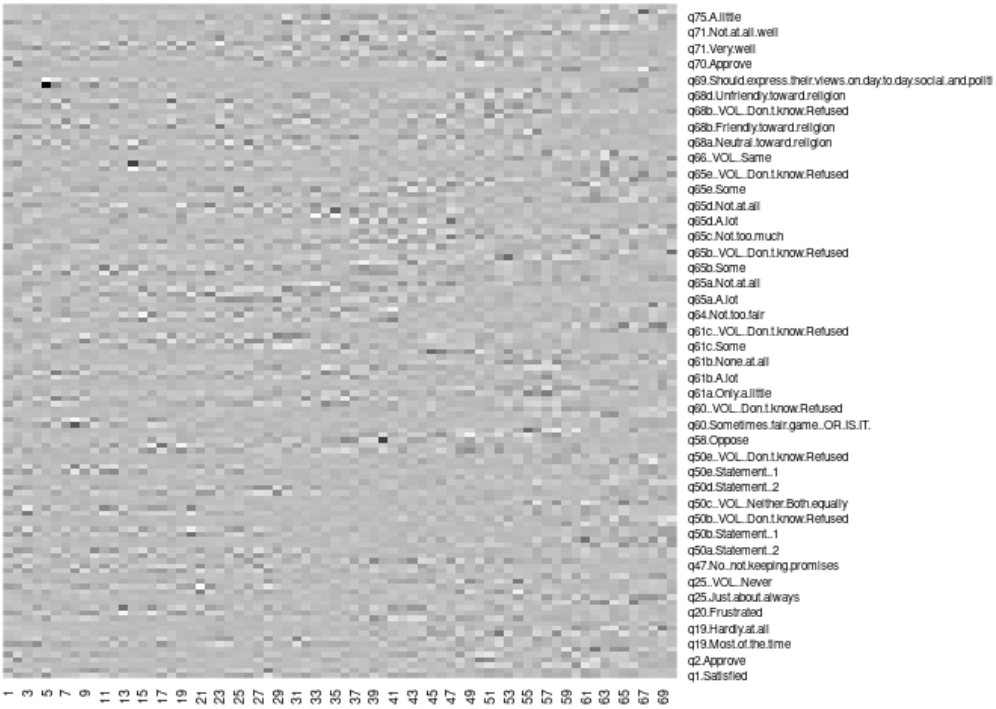


The first components of either PCA or logistic PCA only explain about 13% and 11% of the data variance, respectively. This indicates relatively high independence among the binary features. For downstream analysis, I would want to perverse as much variance as possible even though that means less dimension reduction. To achieve 95% explained variance, standard PCA needs 64 components and logistic PCA needs 70 components. The loadings of these components from each algorithm are shown below as heatmaps.

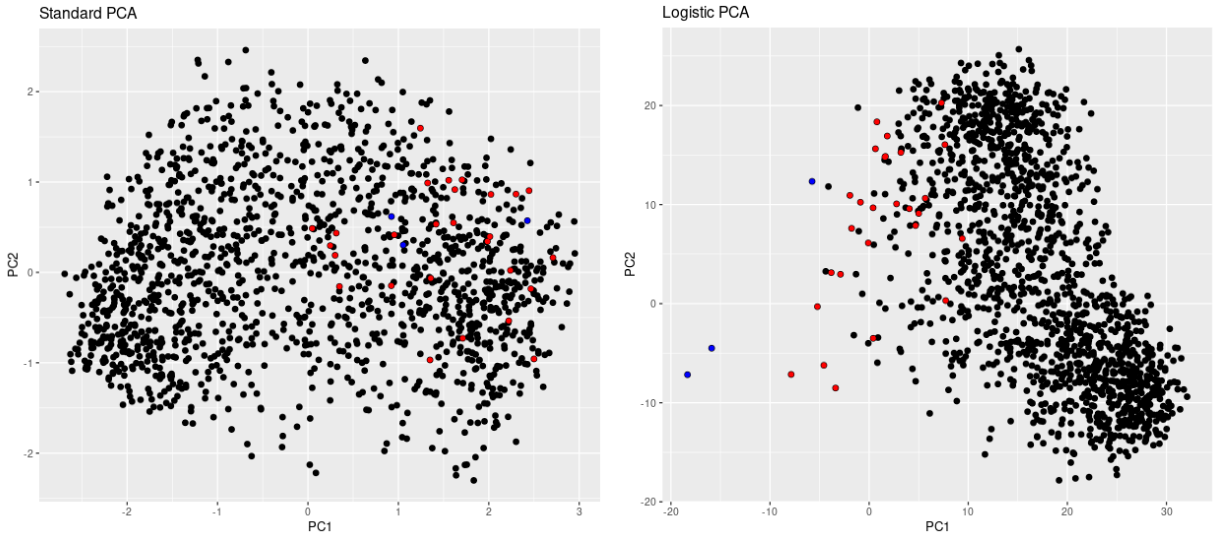
Standard PCA principal component loadings



Logistic PCA principal component loadings

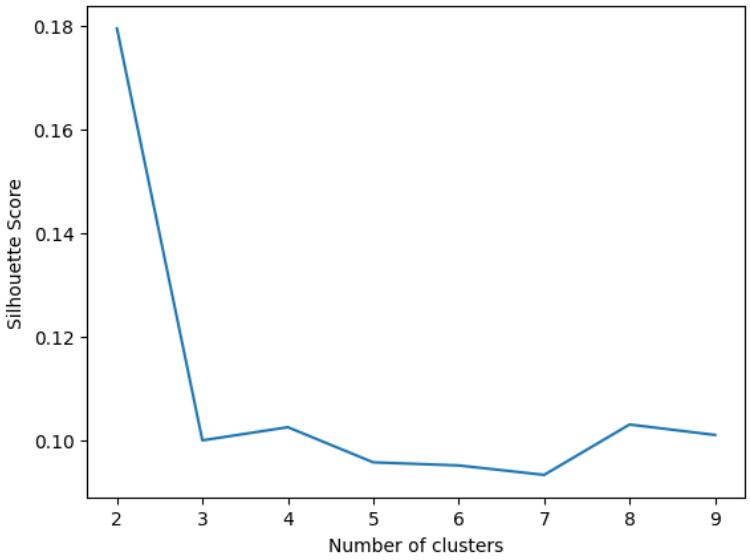


The biplots that show data distribution in the PC1-PC2 space differ greatly for the two algorithms. No obvious cluster can be seen in the data distribution in the standard PCA space, while two clusters seem to be distinguishable by logistic PCA, with some outliers on the left of the plot that deviate from most of the data points. After labeling the top 3 (blue) and top 30 (red) individuals whose survey answers differ the most from the majority, they are indeed the ones deviate more from the rest of the data points in the logistic PC1-PC2 space, but they are indistinguishable from the rest of the data points in the standard PCA space. This suggests that logistic PCA is a better projection method to represent the data structure of this dataset.



3.2 Feature agglomeration

I used Scikit learn framework to perform the remaining sections of the analysis. The Jaccard distance matrix of pairwise features was first computed to perform `FeatureAgglomeration`. Most relevant features (shorter distance) were agglomerated from the bottom up. I chose the linkage parameter as `average` and tested a range of components to fit the data. The Silhouette score plot shown below was used to evaluate the quality of clusters generated by each model.



This indicates that the best clustering model has 2 components, although a 0.18 Silhouette score is still considered low. As the PCA analyses, this result also suggests high independence among the features. Further subgrouping of the independent features will distort the data. Since this method did not generate desirable good quality of clusters, I do not recommend using it for feature engineering of this dataset.

Interestingly though, after inspecting the features in the two feature clusters, I noticed that the second cluster contains mostly "Don't know/Refused" answers to the survey questions (shown below). This suggests some underlying determinant that leads to a similar choice pattern to many survey questions. For instance, centralists might choose these neutral answers (value 1 for these features), while the left and the right are most likely to choose other answers instead (value 0 for these features).

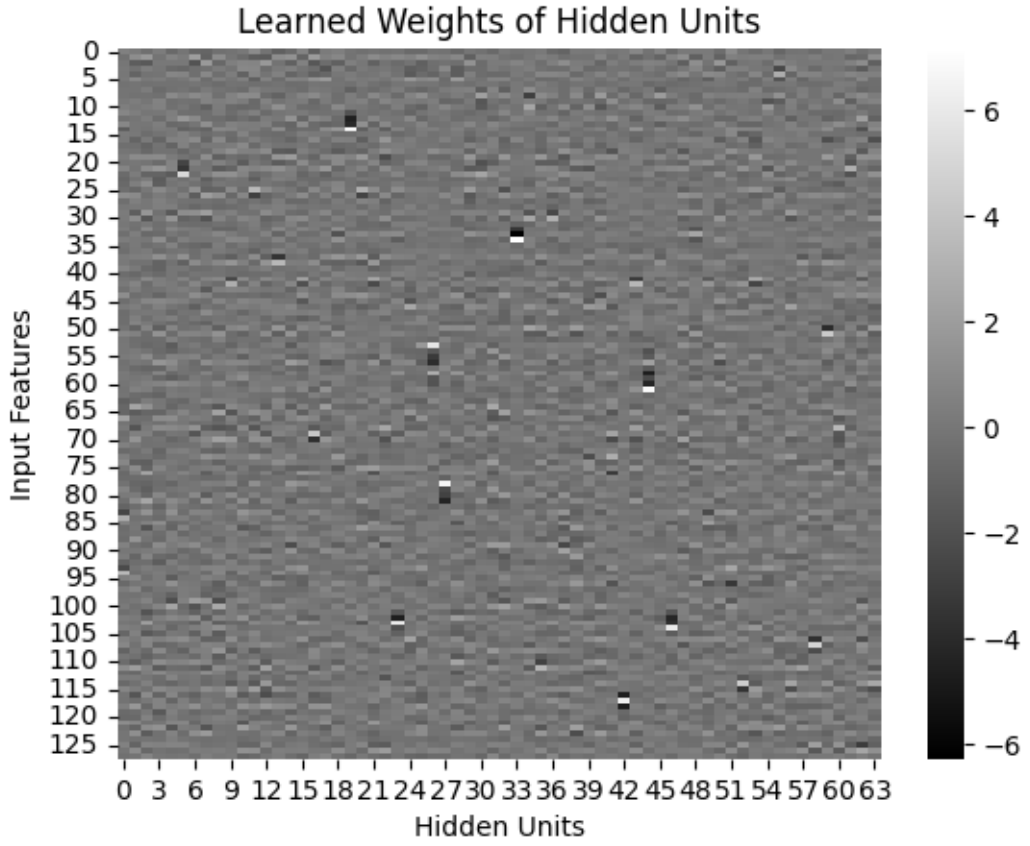
```
np.int64(1): ["q19_(VOL) Don't know/Refused",
 "q20_(VOL) Don't know/Refused",
 "q25_(VOL) Don't know/Refused",
 "q50a_(VOL) Don't know/Refused",
 'q50a_(VOL) Neither/Both equally',
 "q50b_(VOL) Don't know/Refused",
 'q50b_(VOL) Neither/Both equally',
 "q50c_(VOL) Don't know/Refused",
 'q50c_(VOL) Neither/Both equally',
 "q50d_(VOL) Don't know/Refused",
 'q50d_(VOL) Neither/Both equally',
 "q50e_(VOL) Don't know/Refused",
 'q50e_(VOL) Neither/Both equally',
 "q60_(VOL) Don't know/Refused",
 'q60_(VOL) Other/Depends',
 "q61a_(VOL) Don't know/Refused",
 "q61b_(VOL) Don't know/Refused",
 "q61c_(VOL) Don't know/Refused",
 "q64_(VOL) Don't know/Refused",
 "q65a_(VOL) Don't know/Refused",
 "q65b_(VOL) Don't know/Refused",
 "q65c_(VOL) Don't know/Refused",
 "q65d_(VOL) Don't know/Refused",
 "q65e_(VOL) Don't know/Refused",
 "q66_(VOL) Don't know/Refused",
 "q69_(VOL) Don't know/Refused",
 "q75_(VOL) Don't know/Refused"]}
```

3.3 Restricted Bolzmann machine

The algorithm for solving the objective function is Persistent Contrastive Divergence (Tieleman 2008), which is implemented in `BernoulliRBM` in Scikit learn.

Several key hyperparameters for RBMs include number of components (or hidden units that represent the latent variables we are trying to learn), number of iterations (this affects the balance of overfitting and underfitting), and learning rate (this determines how good the algorithm approaches the optima of the objective function). To adjust these hyperparameters, I split the data into an 80% training set and 20% test set. In pure unsupervised learning settings, lack of labels in the training dataset limits gradient descent in backpropagation that fine tunes the weights of an RBM. However, we can still rely on the log likelihood of the visible units in the test set to judge how well the model fits unseen data. The exact likelihood for RBMs is computationally intractable due to the need to sum over all possible configurations of visible and hidden units. Scikit learn offers a function `score_samples` to calculate the pseudo-likelihood of the unseen data. With a parameter grid, I found the highest pseudo-likelihood for the test set has `n_components = 64`, `learning_rate = 0.01`, `n_iter = 500`. Lower learning rate found better optima at higher

iterations. Adaptive learning rate could improve learning. The final receptive fields (or the learned weights of the hidden variables, equivalent to the loadings of the latent variables) are represented below as heatmap:



The RBM transformed data can be used for downstream logistic regression or as inputs into another supervised neural network.

4.0 Discussion and Conclusions

Categorical data generated from political surveys require different statistical learning algorithms other than the ones developed for continuous data. One-hot encoding provides one way to break down the nominal values into binary values for downstream analysis. This process tremendously increases the already high data dimensionality further by several folds, depending on how complex the survey choices are. Dimension reduction techniques decorrelates the features and eliminate noisy features from downstream analysis.

Binary data follows the Bernoulli distribution, which means Euclidean-distance-based projection methods are not suitable for its feature engineering. In this paper, I researched algorithms that can perform dimension reduction with binary data. The comparison of standard PCA and logistic PCA shows that the latter captures the data structure more accurately than the former, even though the explained variances by their first 30 principal components are mostly similar between the two frameworks.

I also tested a tree-based hierarchical clustering method to cluster the features. Although binary tree mimics the feature space partitioning, the feature clusters are of poor quality (low Silhouette

score). However, this analysis did reveal some interesting patterns in selecting neutral answers to multiple survey questions.

Lastly, the RBM neural network based embedding method provides another way to learn latent variables and their loadings, which maintains the model's interpretability the same way as PCA. To compare with logistic PCA, I would need labeled data, such as whether the survey takers identify themselves as a left or a right or a centralist or somewhere in between, or the anonymous voting outcomes. One advantage of a neural network over PCA is that the weights of a neural network can be fine-tuned according to the performance of the downstream analysis.

References

Gower, John C. 1971. "A general coefficient of similarity and some of its properties." *Biometrics* 27 (4): 857–871.

Hinton, G. E. 2002. "Training Products of Experts by Minimizing Contrastive Divergence." *Neural Computation* 14, 1771–1800.

Izenman, Alan Julian. 2008. "Hierarchical Clustering." In *Modern Multivariate Statistical Techniques*, by Alan Julian Izenman, 411-422. New York: Springer.

Jaccard, Paul. 1901. " Étude comparative de la distribution florale dans une portion des Alpes et des Jura." *Bulletin de la Société vaudoise des sciences naturelles (in French)* 37 (142): 547–579.

Landgraf, Andrew J., and Yoonkyung Lee. 2015. "Dimensionality Reduction for Binary Data through the Projection of Natural Parameters." *arXiv* 1510.06112v1.

Pearson, K. 1901. "On lines and planes of closest to systems of points in space." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11), 559-572.

Schalk, Marci, Dean Williams, Stas Kolenikov, and Stas Kolenikov. 2019. *March 2019 Political Survey Methodology Report.* Washington, DC: American Association for Public Opinion Research.

Tieleman, T. 2008. "Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient." *International Conference on Machine Learning (ICML).* Association for Computing Machinery. 1064 - 1071.