Regression Methods in the Fragile Families Challenge

Noah Mandell
Princeton University
nmandell@princeton.edu

Abstract

The Fragile Families and Child Wellbeing Study has developed a dataset characterizing 5,000 children from "fragile families", families that are at greater risk for poverty or other hardship than more traditional families. Recently, the Fragile Families Challenge has been created to encourage the use of data science techniques to facilitate understanding of patterns in the data. This work represents our contribution to the Challenge, in which we develop and train regression methods to predict several key outcomes in the data. We use the ridge regression method and report prediction accuracy metrics for our model.

1 Introduction and Related Work

The Fragile Families and Child Wellbeing Study has developed a dataset characterizing 5,000 children from "fragile families", families that are at greater risk for poverty or other hardship than more traditional families. The study spans 15 years, collecting information about the children, their parents, and their environments. One of the goals of the study is to identify how particular risk factors affect key outcomes for children, such as a child's GPA. The Fragile Families Challenge was created to facilitate understanding of patterns and causality in the study's data by encouraging the blending ideas from social science and data science. In this work, we approach the data from the perspective of machine learning, by training regression models to predict several key outcomes in the data. [1, 8].

With the advent of "big data", data science and machine learning methods are increasingly being used in social science applications. Some other recent endeavors that are combining social science with machine learning include a Stanford study that combines satellite image data with machine learning to map impoverished regions in Africa [3]; a study that uses aggregate data from mobile phones and geospatial data to measure and target poverty without having to rely on census-like data [10]; and a study of how machine learning can be used to improve and understand the decision-making processes of a judge's decisions in court [4].

2 Methods

2.1 Data Description

The raw dataset is represented as an array with 4242 rows and 11988 columns. Each of the rows corresponds to a child in the study. Each of the columns corresponds to a response to a survey question, or in some cases, a response constructed after-the-fact from other responses. A large portion of the data is encoded with integers, such as 1 = No and 2 = Yes; this numerical encoding results in categorical variables. There are also continuous variables, such as income and various ages. [5]

For each row of data there are six Year 15 outcomes that we would like to predict [5]. Three of the outcomes are continuous variables:

- **GPA**: The child's self-reported GPA on a 4 point scale.
- **Grit**: A measure of the child's passion and perseverance, based on answers to 4 questions. Ranges from 1 to 4.
- **Material hardship**: A measure of extreme poverty, based on answers to 11 questions. Ranges from 0 to 1.

The remaining three Year 15 outcomes are binary:

- **Eviction**: Whether child's primary caregiver was evicted from home in time since Year 9 survey.
- **Layoff**: Whether child's primary caregiver was laid off in time since Year 9 survey.
- **Job Training**: Whether child's primary caregiver took any classes to improve job skills since Year 9 survey.

Note that some of the outcomes are missing for some rows; we ignore these cases when training our models.

2.2 Data Processing, Imputation, and Encoding

For a dataset as large and raw as this one, how the data is processed and cleaned can be crucial. We first note that there are some columns which are completely filled with NA values; these correspond to features which have been redacted, so we immediately remove these columns. We also note that while most of the data is numeric, some of the data is in the form of text strings. For simplicity, we drop all columns which contain any strings. We also remove columns with zero variance, since this data will not be predictive. These steps alone eliminate over 1,500 features, which is roughly 13% of the total number of features.

A significant challenge in this dataset is the fact that a large portion (roughly 67%) of the data is missing. Missing data is generally encoded with negative integers. In many cases the data is missing for valid reasons; a common example is a 'valid skip' (encoded as -6), where a question was intentionally not asked based on previous responses. Data can also be missing for invalid reasons, such as when one of the interview subjects was not able to do an interview in a given round of interviews (encoded as -9), or when the interviewer incorrectly omitted a question (-3 or -5). [5]

Imputation must therefore be used to fill in the missing values. We will take the simple approach of imputing by replacing missing values with the most frequent value in the column, but with a few caveats. First, before imputing we drop columns with more than 30% of the data missing. This reduces the number of columns significantly, but it also reduces the ratio of missing to non-missing data, which is the goal of this step. We choose the threshold of 30% missing per column because this produces a dataset with only 20% of the data missing across all columns. The result of this processing is a reduced dataset with about 1800 columns.

We next attempt to distinguish the categorical variables from the continuous ones. We do this by taking advantage of the fact that the numerical coding for the categorical variables uses only integers, and (as far as we can tell) does not make use of values in the range 20-99. We also note that variables that take on values exclusively in the range 1940-2016 are likely to represent years and thus are continuous. Thus we use these criteria to label variables as likely categorical and likely continuous. This results in nearly 90% of the variables being labeled categorical. This likely over-predicts the number of categorical variables, but this is preferable to under-predicting the number of categorical variables; see below.

We then transform all categorical data to a sparse representation using a one-hot encoding. This transforms each categorical variable with m distinct values into m binary features, where the i^{th} binary feature is active only when the original variable took on the i^{th} value. This representation is required for regression with categorical variables. The data represented by continuous variables is left unchanged, except for normalization in some cases. The final step of data processing is to filter out one-hot-encoded columns with low variance. We choose the variance threshold so that categorical responses with a frequency of less than 100 are dropped. The final data representation has 4341 columns, of which most columns are sparse and binary.

Note that the consequences for one-hot encoding a non-categorical variable is only loss of information about the ordering of the values. Compared to the consequences of not one-hot encoding a categorical variable, which would effectively impose incorrect assumptions about ordering of the values, we see that over-predicting the number of categorical variables is indeed preferable, as stated above.

2.3 Regression Methods

We use regression methods to predict the six outcomes, both continuous and binary. For the binary outcomes, our models effectively predict the probability (between 0 and 1) of a binary outcome being true. We investigated several regression methods using the Python SciKitLearn libraries [7], including Ordinary Least Squares regression, Ridge regression, Huber regression, and Random Forest regression. In the interest of brevity, we present results only from our Ridge regression model, which we found to be most successful for predicting the desired outcomes. See section 3 for a detailed description of the Ridge regression method.

As part of our model fitting procedure, we use efficient leave-one-out cross-validation to choose the regularization parameter for our Ridge regression model [9], resulting in typical values of $\alpha \sim 15 - 20$. Since this cross-validation is included in the model fitting procedure, it thus becomes an inner cross-validation loop when using an outer cross-validation to evaluate model performance. This nested cross-validation is necessary to avoid biasing performance estimates [2].

2.4 Evaluation and Predicting Performance

We use two evaluation metrics: mean squared error (MSE) and R^2 score (R2). These are defined by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$$

where $\bar{y}$ is the mean of y. Note that MSE is the evaluation metric used for the FFC leaderboard. Our models are fit to minimize MSE. A better fit also tends to increase R^2 (towards 1).

We do not have access to the correct outcomes for the testing set, so in order to predict the performance of our models on the testing set, we use a bootstrap method on the training set. From the entire training dataset, we randomly select 50 different bootstrap samples each composed of 90% of the training data. For each of these samples, we train our model and then predict the outcomes of the remaining 10% of the training data. Repeating this 50 times with randomly selected splits (with replacement), we can get a good estimate of the metrics by taking the mean of the metrics across the 50 bootstrap samples, as well as a 95% confidence interval by taking the standard deviation.

3 Ridge Regression

Ridge regression differs from ordinary least squares regression in that the fitting parameters are regularized. For a dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2)...(\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x} \in \mathbb{R}^d$ are the d-dim input feature vectors and $y \in \mathbb{R}$ are the true outcomes, we assume a linear model of the form

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \epsilon = \hat{y} + \epsilon,$$

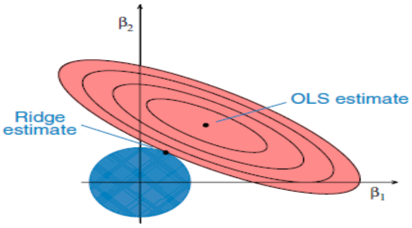


Figure 1: Geometric interpretation of ridge regression. Source: [11]

where $\boldsymbol{w} \in \mathbb{R}$ are the model parameters, $\hat{y}$ are the predicted outcomes, and $\epsilon = y - \hat{y}$ is the difference between the true and predicted outcomes. Assuming that ϵ is Gaussian with fixed noise, we can express the model as a conditional probability:

$$p(y|\boldsymbol{x}, \boldsymbol{w}, \sigma^2) = \mathcal{N}(y|\boldsymbol{w}^T\boldsymbol{x}, \sigma^2)$$

where $\mathcal{N}(y|\mu, \sigma^2)$ represents a normal (Gaussian) distribution of y with mean μ and variance σ^2.

As in ordinary least squares, we fit the model parameters $\boldsymbol{w}$ by maximizing the log posterior distribution (MAP). The difference in ridge regression is that we include a zero-mean Gaussian prior on the model parameters, $p(\boldsymbol{w}) = \prod_i \mathcal{N}(w_i|0, \tau^2)$. The resulting MAP problem is then given by

$$\arg\max_{\boldsymbol{w}} \sum_{i=1}^{N} \log \mathcal{N}(y_i|w_0 + \boldsymbol{w}^T\boldsymbol{x}_i, \sigma^2) + \sum_{j=1}^{p} \log \mathcal{N}(w_j|0, \tau^2).$$

We can show that this is equivalent to

$$\arg\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - (w_0 + \boldsymbol{w}^T\boldsymbol{x}))^2 + \alpha||\boldsymbol{w}||_2^2$$

Note that the first term is the same as in ordinary least squares. The second term is a regularization or penalty term, with $\alpha \doteq \sigma^2/\tau^2$ the regularization parameter. It uses the ℓ_2 norm, which encourages the model parameters to be small and helps to prevent overfitting.

The MAP problem can be solved analytically, just as in ordinary least squares. The solution is given by

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\mathbf{I})^{-1}\boldsymbol{X}^T\mathbf{y}.$$

Again, we see that this solution is similar to the ordinary least squares solution, with the exception of the term involving α. This term actually helps numerically, since adding a small value α to the diagonal makes a matrix better conditioned and therefore more likely to invertible. [6]

4 Results

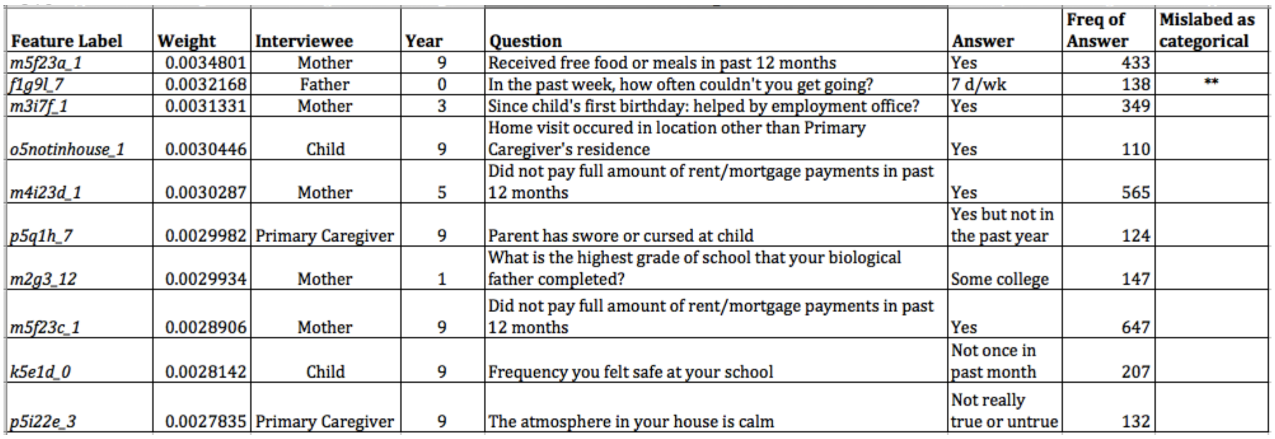
4.1 Evaluation results

As described above, we use a bootstrap method to estimate the mean squared error and R^2 metrics from the training data for each of the six predicted outcomes. These estimates, along with 95% confidence intervals, are given in the "Boot" columns of Table 1. We also obtain the mean squared error on the testing set from the FFC submission platform (username *nmandell*), which is given in the "FFC" column of Table 1. We can see that for all six outcomes the actual mean squared error from FFC is within the confidence interval of the bootstrap MSE predictions, showing that the bootstrap method is indeed effective at predicting the performance of the models.

	FFC MSE	Boot MSE	Boot R2
GPA	0.3797	0.3795 ± 0.8222	0.1254 ± 0.1183
Grit	0.21443	0.2299 ± 0.0475	0.0293 ± 0.0694
Material Hardship	0.02524	0.0201 ± 0.0062	0.1675 ± 0.1132
Eviction	0.05153	0.0548 ± 0.0289	0.0405 ± 0.062
Layoff	0.17358	0.1698 ± 0.0391	-0.0095 ± 0.0513
Job Training	0.20282	0.1725 ± 0.0331	0.0399 ± 0.0569

Table 1: Results for each of the six outcomes. "FFC" is from FFC leaderboard, evaluated on testing set. "Boot" is from bootstrapped estimates on training set.

It appears that the model for material hardship is most successful, as it has both the lowest MSE and the largest R^2 values. Table 2 shows the top ten most predictive features for this outcome. Looking at the questions related to these features, we can logically see that information such as whether or not the mother received free meals, or whether or not the mother was unable to make full payment for rent or mortgage, is quite relevant to the idea of material hardship. This kind of predictive information is certainly valuable to the creators of the Fragile Families study and policy makers



Feature Label	Weight	Interviewee	Year	Question	Answer	Freq of Answer	Mislabed as categorical
m5f23a_1	0.0034801	Mother	9	Received free food or meals in past 12 months	Yes	433	
f1g9l_7	0.0032168	Father	0	In the past week, how often couldn't you get going?	7 d/wk	138	**
m3i7f_1	0.0031331	Mother	3	Since child's first birthday: helped by employment office?	Yes	349	
o5notinhouse_1	0.0030446	Child	9	Home visit occured in location other than Primary Caregiver's residence	Yes	110	
m4i23d_1	0.0030287	Mother	5	Did not pay full amount of rent/mortgage payments in past 12 months	Yes	565	
p5q1h_7	0.0029982	Primary Caregiver	9	Parent has swore or cursed at child	Yes but not in the past year	124	
m2g3_12	0.0029934	Mother	1	What is the highest grade of school that your biological father completed?	Some college	147	
m5f23c_1	0.0028906	Mother	9	Did not pay full amount of rent/mortgage payments in past 12 months	Yes	647	
k5e1d_0	0.0028142	Child	9	Frequency you felt safe at your school	Not once in past month	207	
p5i22e_3	0.0027835	Primary Caregiver	9	The atmosphere in your house is calm	Not really true or untrue	132	

Table 2: Top 10 most predictive features for the material hardship outcome.

in general. We also note that all of the top features are ones that have been labeled (sometimes incorrectly) as categorical variables. The top continuous variable is ranked number 1204, *p5h15c*, which measures the number of hours the child spends with someone who is smoking. It is unclear why our model seems to prefer the categorical variables, other than the fact that 90% of the data is categorical.

The challenges of making predictions from this kind of real-life dataset are shown by the near-zero R^2 values. A value of $R^2 = 0$ would indicate that random guessing should lead to better predictions, so our models are not significantly better than random guessing. Much more complex models or more training data may be required to get R^2 values nearer to 1. However by comparing our scores to others on the FFC leaderboard, we see that our model appears to be competitive with other groups' models, ranking in the 25th percentile for all six outcomes.

4.2 Computational Speed

The computation time is dominated by the bootstrapping procedure, which takes 1-2 minutes per outcome on 2 cores. Fitting the ridge regression model (including cross-validated hyperparameter selection) and making predictions for the entire training set takes only about 1-2 second for each outcome. Note that ridge regression was by far the most efficient of all the regression methods we tried, which was another reason for choosing this method for our analysis.

5 Discussion and Conclusion

In this work, we used regression models to predict six key outcomes for at-risk children using data from the Fragile Families and Child Wellbeing Study. Starting from a feature set with nearly 12000 features, we used a number of techniques to filter the data, impute missing values, and encode the data in a way that enables training with regression models. We found a ridge regression model to be most successful for predicting the outcomes, and reported prediction metrics via bootstrapping. We also showed that features identified by the model as most predictive appear to be logical.

There are a number of extensions that could be made to improve the performance of our models, but one that might be important is using more sophisticated imputation strategies for missing data. Simply imputing with the mean or mode, as we have, is likely to degrade performance compared to more sophisticated imputation alternatives. One could also investigate more complex regression models, especially models that account for the latent-in-time structure of the data, with data coming from several years of interviews. Finally, one could combine several somewhat successful models into a single super-model; this is one of the intended goals of the Fragile Families Challenge contest and submission platform.

References

[1] Fragile families challenge website. http://www.fragilefamilieschallenge.org/, 2017.

[2] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

[3] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[4] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. Technical report, National Bureau of Economic Research, 2017.

[5] Ian Lundberg. Fragile families challenge blog. http://www.fragilefamilieschallenge.org/blog/, 2017.

[6] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Nancy E Reichman, Julien O Teitler, Irwin Garfinkel, and Sara S McLanahan. Fragile families: Sample and design. *Children and Youth Services Review*, 23(4-5):303–326, 2001.

[9] Ryan M Rifkin and Ross A Lippert. Notes on regularized least squares. 2007.

[10] Jessica E Steele, Pål Roe Sundsøy, Carla Pezzulo, Victor A Alegana, Tomas J Bird, Joshua Blumenstock, Johannes Bjelland, Kenth Engø-Monsen, Yves-Alexandre de Montjoye, Asif M Iqbal, et al. Mapping poverty using mobile phone and satellite data. *Journal of The Royal Society Interface*, 14(127):20160690, 2017.

[11] STAT897D Penn State University. *Ridge Regression*, 2016 (accessed Feb. 28, 2017). https://onlinecourses.science.psu.edu/stat857/node/155.