PS1 Solution

January 10, 2019

MACS 30150
Li Liu

1. Classify a model from a journal

(a) I find a statistical model from the article "Machine learning methods for demand estimation" published at *American Economic Review*.

(b) Bajari, P., D. Nekipelov, S.P. Ryan, and M. Yang. (2015). Machine learning methods for demand estimation. *American Economic Review*, 105 (5), 481-485.

(c)

$$Y = \alpha + \alpha_1 y_{Linear} + \alpha_2 y_{Stepwise} + \alpha_3 y_{Forward} + \alpha_4 y_{LASSO} + \alpha_5 y_{RandomForest} + \alpha_6 y_{SVM} + \alpha_7 y_{Bagging} + \alpha_8 y_{Logit} + \epsilon_i \quad (1)$$

(d)

Endogenous variable: Y is the combined and weighted predictted vlaue of log quantity sold per week('demand').

Exogenous variables: y are the predictted log sales vlaue using each of the eight different statistical models (Linear, Stepwise, Forward stagewise, LASSO, Random Forest, SVM, Bagging, Logit).

(e)

The model is static, linear, and deterministic.

(f)

Since the data contains over 1 million observations, the model could also include $\alpha_9 y_{NeuralNets}$, where $y_{NeuralNets}$ is the predictted value using neural networks method.

2. Make your own model (a-c)

A simplied logistic regression model for people to make marriage decision:

$$Y = \begin{cases} 1 = \text{get married}, & \text{if } p(X) >= 0.5 \\ 0 = \text{not get married}, & \text{if } p(X) < 0.5 \end{cases} \tag{2}$$

where

$$p(X) = \frac{\exp(\beta_0 + \beta_1 \text{Age} + \beta_2 \text{Career} + \beta_3 \text{Characters} + \beta_4 \text{ParentsSupport})}{1 + \exp(\beta_0 + \beta_1 \text{Age} + \beta_2 \text{Career} + \beta_3 \text{Characters} + \beta_4 \text{ParentsSupport})} \tag{3}$$

In the specification of *p(X)*:

`Age` is continuous. People are more eager to get married as they get older.

`Career` is ordinal. It is classified into five categories based on job stability, income level, promotion prospect, position prestige, etc. 1 is the worst and 5 is the best.

`Characters` is ordinal. It denotes whether the future couple share the same characters. This variable summaries the factors such as religions, culture, education, etc. 1 means the two disagree with many things, 10 means the two appreciate each other in life.

`ParentsSupport` is binary. 1 means parents from both sides approve the marriage, and 0 otherwise.

(d) Key factors that influence this outcome are `career` and `characters`. I think people are more prepared for the marriage and life-long committment if they have the ability to raise the family financially and communicate with the spouse joyfully.

(e) `ParentsSupport`, `Age` ,`career` and `characters` capture most of the factors for marriage decision. I chose them based on both sociological research and observations from newly married acquaintances.

(f) A preliminary test to examine the significance of factors could be estimating the parameters by maxmimizing the likelihood function using the real data from Census Bureau and government's marriage record.

Extra Play around the model with simulated data to evaluate my current marriage probability!

```
In [120]: import numpy as np
          import pandas as pd
          import statsmodels.api as sm
          import matplotlib.pyplot as plt
          import warnings
          warnings.filterwarnings("ignore")

In [150]: Age=np.random.normal(30,3,100)
          Career=np.random.normal(4,0.5,100)
          Characters=np.random.normal(7,0.5,100)
          ParentsSupport=np.random.choice([0, 1], size=100, p=[.3, .7])
          Outcome=np.random.choice([0, 1], size=100, p=[.4, .6])
          Marriage=pd.DataFrame({'Age':Age.astype(int),'Career':Career.astype(int),
                                 'Characters':Characters.astype(int),
```

```
                                'ParentsSupport':ParentsSupport,
                               'Outcome':Outcome})
          Marriage['const'] = 1
          Marriage.head(5)
```

Out[150]:

	Age	Career	Characters	ParentsSupport	Outcome	const
0	33	4	5	1	0	1
1	28	4	6	1	1	1
2	27	3	7	1	1	1
3	31	4	6	1	0	1
4	30	3	6	1	1	1

```
In [151]: X, y = sm.add_constant(Marriage[['Age','Career','Characters','ParentsSupport']],
                                 prepend=False), Marriage['Outcome']
          m2= sm.Logit(y, X).fit()
          print(m2.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.689136
         Iterations 4
                         Results: Logit
=================================================================
Model:              Logit            No. Iterations:   4.0000
Dependent Variable: Outcome          Pseudo R-squared: 0.003
Date:               2019-01-08 16:09 AIC:              147.8271
No. Observations:   100              BIC:              160.8530
Df Model:           4                Log-Likelihood:   -68.914
Df Residuals:       95               LL-Null:          -69.135
Converged:          1.0000           Scale:            1.0000
-----------------------------------------------------------------
                  Coef.  Std.Err.    z    P>|z|   [0.025 0.975]
-----------------------------------------------------------------
Age               0.0081   0.0665  0.1221 0.9028 -0.1222 0.1384
Career            0.1813   0.3281  0.5527 0.5804 -0.4616 0.8243
Characters        0.0802   0.3347  0.2396 0.8107 -0.5758 0.7361
ParentsSupport    0.0416   0.4817  0.0865 0.9311 -0.9024 0.9857
const            -1.3014   2.9215 -0.4454 0.6560 -7.0275 4.4247
=================================================================
```

```
In [161]: prob=np.sum(np.array(m2.params)*[23,2,7,1,1])
          print("Outcome for you:",1 if np.exp(prob)/(1+np.exp(prob))>0.5 else 0 )
          print("No matter how sloppy the model is, I shouldn't get married now :)")
```

```
Outcome for you: 0
No matter how sloppy the model is, I shouldn't get married now :)
```

Hypothetically, most of the parameters should be positive. If that's the case, in order to increase my confidence of marriage decision, I don't need to hurry up at the 20s. I could increase `career` factor by landing a good career after graduate school, find someone who has high `Characters` score, and then get `ParentsSupport`!