DEMAND ESTIMATION WITH MACHINE LEARNING

Georg Zhelev

University of Hamburg

georg.zhelev@gmail.com

January 24, 2021

Overview

Introduction: Machine Learning & Goal of Research

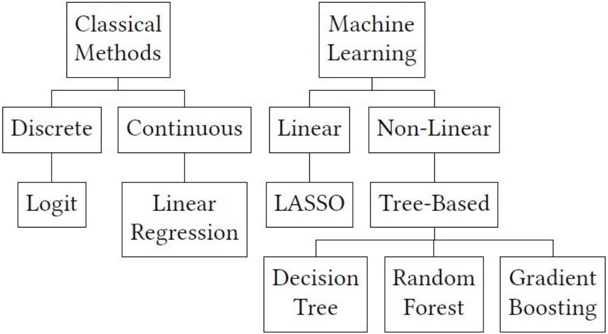
- **Motivation**: to enable techlogical services data must be **stored** and **analysed**
- **Definition**: Advanced statistical prediction techniques suitable to analyze **large** amounts of data
- **Specific area**: Demand Estimation in a discrete choice setting
- **Problem**: Standard statistical methods **fail**, when the number of features is very large compared to the number of observations
- **Goal of Research** empirical application of classical and machine learning methods
- **simulated** product data
- **performance** is measured and the the number of features is **varied**

Theoretical Framework: Variables that Influence Demand

$$D_a = f(X_1, ..., X_n)$$

1. Price
2. Quality
3. Popularity

Goal: transform the general function f into a specific one: Determine to what extent variable affects demand.



Theoretical Framework: Utility Model

- customer n faces a buying decision among a set of competing alternatives j
- a person obtains a net benefit U from choosing one product j
- β is a coefficient of how observed factors x contribute to U
- ϵ captures the factors that are included in U, but are not observed
- ϵ has an extreme value distribution (heterogenous individuals)

Theorem (Utility Model)

$$U_{nj} = \beta * x_{nj} + \epsilon_{nj};\ \forall j,$$

Theorem (Utility Model Concise)

$$U_{nj} = V_{nj} + \epsilon_{nj},$$

Methodology: Simulation

Assign Indexes

$$n = 1, ..., N, \quad p = 1, ..., P, j = 1, 2, 3 \,.$$

Assign Coefficients

$$\begin{bmatrix} -5 & 3 & 5 & 0 & 0 & \dots & \beta_p \end{bmatrix} .$$

Generate Product Characteristic-Means μ and name columns

Alternative	Alternative Name	μ Price	μ Quality	μ Popularity
j=1	TF (Tiger Flakes)	3.90	5	2
j=2	HB (Honey Bits)	3,50	1	4
j=3	NC (Nougat Crisps)	1.50	1	2

Methodology: Simulation

Generate X-Variables using means and diagonal covariance structure

$$\begin{bmatrix} 3.9 & 5.0 & 2.0 & 0 & 0 & \dots & \mu_p \end{bmatrix},$$

$$\begin{bmatrix} 0.2 & 0 & \dots & 0 \\ 0 & 0.2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0.2 \end{bmatrix}^{1} \rightarrow \begin{bmatrix} x_{1,1}^{(1)} & x_{1,2}^{(1)} & \dots & x_{1,p}^{(1)} \\ x_{2,1}^{(1)} & x_{2,2}^{(1)} & \dots & x_{2,p}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1}^{(1)} & x_{n,2}^{(1)} & \dots & x_{n,p}^{(1)} \end{bmatrix}$$

Calculate Representative Utility using $V_{n,j} = \beta_p * t(X_{n,p})$,

$$\begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,1} & v_{2,2} & v_{2,3} \\ \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & v_{n,3} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} * \begin{bmatrix} x_{1,1}^{j} & x_{1,2}^{j} & \dots & x_{1,p}^{j} \\ x_{2,1}^{j} & x_{2,2}^{j} & \dots & x_{2,p}^{j} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1}^{j} & x_{n,2}^{j} & \dots & x_{n,p}^{j} \end{bmatrix}^{T}$$

[1] diagonal matrix with variance $\sigma = 0.2$. Generated X's vary around means

Methodology: Simulation

Summary Statistics of Generated Utilities

V1	V2	V3
Min. :-6.176	Min. :-3.674	Min. :-3.395
Median : 5.508	Median : 6.087	Median : 5.802
Mean : 5.759	Mean : 5.718	Mean : 5.749
Max. :15.053	Max. :16.076	Max. :13.755

Generate a $n * j$ matrix of the random error ϵ

$$\epsilon \sim \text{Gumbel}(0, 1.64)$$

Variance satisfies the extreme value distribution-assumption. Signal-to-Noise Ratio of 2-3.

Add Random Error to Utility

$$\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ \vdots & \vdots & \vdots \\ u_{n,1} & u_{n,2} & u_{n,3} \end{bmatrix} = \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,1} & v_{2,2} & v_{2,3} \\ \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & v_{n,3} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,1} & \epsilon_{1,2} & \epsilon_{1,3} \\ \epsilon_{2,1} & \epsilon_{2,2} & \epsilon_{2,3} \\ \vdots & \vdots & \vdots \\ \epsilon_{n,1} & \epsilon_{n,2} & \epsilon_{n,3} \end{bmatrix}.$$

Methodology: Simulation

Generate Response Variable y

$$\sum_{i=1}^{J} P_{ni} = \sum_{i=1}^{J} \frac{e^{U_{ni}}}{\sum_j e^{U_{nj}}} = 1\,.$$

$$0 \leq P_{n,j} \leq 1.$$

$$\begin{bmatrix} 0.05 & 0.95 & 0.00 \\ 0.01 & 0.00 & 0.98 \\ 0.23 & 0.32 & 0.45 \\ \vdots & \vdots & \vdots \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 3 \\ 3 \\ \vdots \\ y_n \end{bmatrix} \rightarrow \begin{bmatrix} HB \\ NC \\ NC \\ \vdots \\ y_n \end{bmatrix}$$

Tabulation of y_n (n=100)

TF	HB	NC
31	34	35

Methodology: Training, Making Predictions, Assessing Performance

Safe Data to a Data Frame

$$\begin{bmatrix} HB & x1.TF_1 & \dots & x10.TF_1 & x1.HB_1 & \dots & x10.HB_1 & x1.NC_1 & \dots & x10.NC_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ TF & x1.TF_n & \dots & x10.TF_n & x1.HB_n & \dots & x10.HB_n & x1.NC_n & \dots & x10.NC_n \end{bmatrix},$$

1. Data Split (Train/Test: 50/50)
2. Pre-Process (One-vs-rest Approach)
3. Train (estimate β's)
4. Make Predictions (predict $\hat{y}$)
5. Assess Performance $\left(\frac{\#correct}{all\ predictions}\right)$

observations	$n = 300$
features	$p = (10, 50, 150, 290)$
coefficients	$\beta_1 = -5,\ \beta_2 = 3,$ $\beta_3 = 5,\ \beta_{rest} = 0$
run time	2 Minutes

Results: Prediction Accuracy all Models (in %)

	Train	Test
Logit	76	78
Linear Regression	76	71
Lasso	72	77
Decision Tree	67	49
Random Forest	100	63
Gradient Boosting	88	61

Table: p=10

	Train	Test
Logit	77	62
Linear Regression	72	45
Lasso	62	54
Decision Tree	58	41
Random Forest	100	45
Gradient Boosting	100	45

Table: p=50

	Train	Test
Logit	100	41
Linear Regression	80	33
Lasso	70	65
Decision Tree	69	47
Random Forest	100	51
Gradient Boosting	100	49

Table: p=150

	Train	Test
Logit	100	36
Linear Regression	84	29
Lasso	74	64
Decision Tree	65	47
Random Forest	100	50
Gradient Boosting	100	47

Table: p=290

Results: Coefficients

Table: Relevant Coefficients

	p=10	p=50	p=150	p=290
X1	-3.1	-3.4	-28.9	-12.3
X2	1.8	1.9	16.5	6.5
X3	3.1	3.1	31.8	11.2

Table: Logit

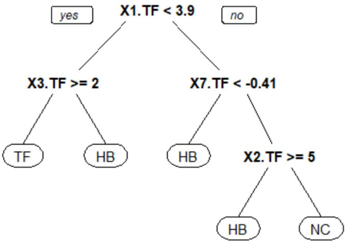
	p=10	p=50	p=150	p=290
X1.TF	-3.9	-4.3	-10.0	-677.7
X2.TF	1.4	1.5	4.6	634.9
X3.TF	2.1	2.9	6.1	-386.8

Table: Linear Regression

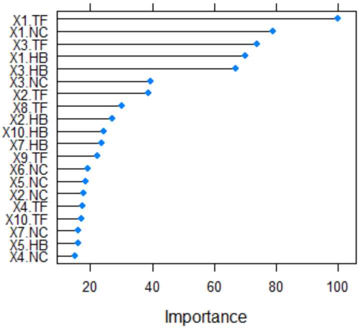
	p=10	p=50	p=150	p=290
X1.TF	-3.9	-3.8	-4.2	-3.5
X2.TF	1.4	0.7	1.4	0.7
X3.TF	2.1	3.6	2.7	3.0

Table: Lasso

Results: Variable Splits of Tree Based Methods



(a) Tree Split p=10



(b) Random Forest Variable Importance p=10

Discussion & Conclusion

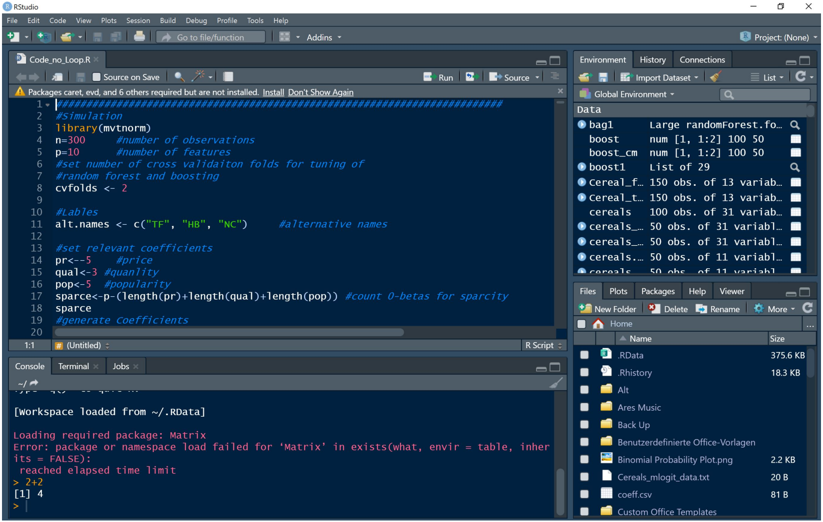
- **Hypothesis 1**: discrete choice model (Logit) is good for a small number of features p, but starts to overfit as $p \Rightarrow n$
- **Hypothesis 2**: ML methods perform better than classical methods, when $p \Rightarrow n$
- **Hypothesis 3**: Linear models will have a better prediction accuracy as non-linear models
- **Hypothesis 4**: A linear model, which practices variable selection will outperform a linear model which doesn't

The End

[1] Link to Paper and Code:
https://gzhelev2020.github.io/portfolio/01.thesis/

Appendix

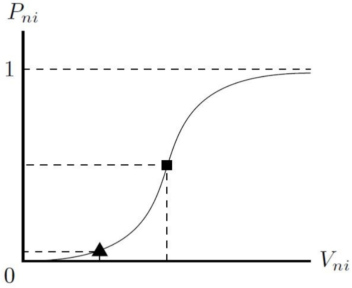
Appendix: Work-Environment



Appendix: Logit Model

$$Prob(U_{ni} > U_{nj}) \qquad P_{ni} = \frac{e^{U_{ni}}}{\sum_j e^{U_{nj}}}$$

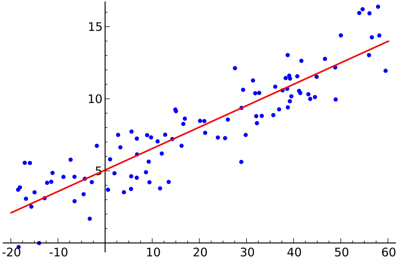
$$0 < P_i < 1 \qquad \sum_{i=1} P_i = 1$$



- Estimation: choose β's so the probabilty of reporoducing the observed model is maximized

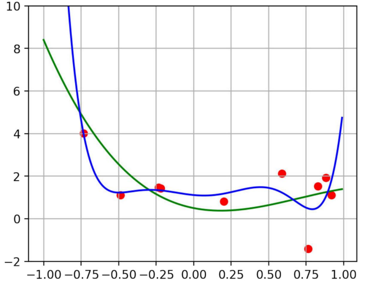
Appendix: Linear Regression

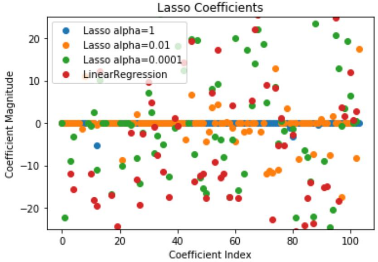
- The ordinary least squares, estimates β that $\min_{\beta \in \mathbb{R}} \sum_i (Y_i - \beta' X_i)^2$
- as $p \to n$, coefficients β not accurately estimated
- Sparsity: only subset of p are important, can apply Lasso



Applied Methods: Lasso Regression and Overfitting

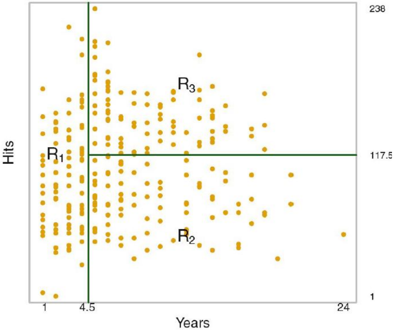
- the lasso estimator is $\min_{\beta \in \mathbb{R}} \sum_i (Y_i - \beta' X_i)^2 + \alpha * \sum_{j=1}^{p} |\beta_j|$
- penalty/regularization term adds to the loss
- all coefficients are reduced in direction of zero
- ensures lasso does not overfit

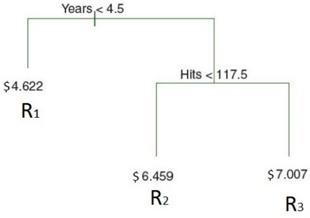




Appendix: Decision Tree, Random Forest and Gradient Boosting

- Partition characteristic space into regions
- chooses best features on which to divide the characteristic space
- predicted response of an observation by the mean response of the training observations





Appendix: Model Characteristics

observations	$n = 300$
features	$p = (10, 50, 150, 290)$
assigned coefficients	$\beta_1 = -5,\ \beta_2 = 3,\ \beta_3 = 5,\ \beta_{rest} = 0$
total run time	2 Minutes

Table: Signal-to-Noise Ratio

	Signal-to-Noise	Noise
p=10	2.24	1.64
p=50	3.02	1.64
p=150	2.43	1.64
p=290	2.84	1.64