Penalized Logistic Regression

By:
[redacted] Ethan Scott (30117295), [redacted]

Background - Standard Logistic Regression

- The model:

$$y \sim B(1, p_x) \qquad\qquad p_x = expit(\beta_0 + \sum_{i=1}^{n} \beta_i x_i)$$

- Want to **estimate** the “β” parameters

- Previously looked at the **MLE**
    - Best choice of parameters maximizes the **likelihood function**

- For practicality we look at the **log-likelihood function**

$$L^* = \prod_{i=1}^{n} P(D_i | X = x_i) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i} \qquad\qquad \log(L^*) = \sum_{i=1}^{n} y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)$$

Logistic Regression Setbacks

- May lead to **overfitting** (when data is **sparse**)
  - Becomes degenerate if there are more features than samples
- Does not **incorporate prior knowledge** about coefficients [1]
  - Coefficients should be “reasonably” small
  - Want to **penalize** large coefficients
- Toy example on the right
  - 10 input dimensions
  - 10 Bernoulli samples

	Actual	Standard	Ridge	Lasso
(Intercept)	-0.6264538	-15.326573	-0.6923376	-0.6212066
X1	0.1836433	25.164221	0.5581994	0.4283797
X2	-0.8356286	-11.936938	-0.1678179	.
X3	1.5952808	28.645658	1.3956100	1.9231301
X4	0.3295078	48.602729	1.4190559	1.4399812
X5	-0.8204684	16.008094	-0.1138861	.
X6	0.4874291	-9.499460	0.8503816	0.6923992
X7	0.7383247	5.353122	1.3408711	1.2513540
X8	0.5757814	-6.708295	0.4025312	.
X9	-0.3053884	14.789231	-0.3127941	.

What is Penalized Logistic Regression?

- Rephrase the problem in terms of **minimizing a cost function**

$$C(\beta, x, y) = -\log(L^*)$$ [2]

- We add a **penalty for large coefficients**

$$C'(\beta, x, y) = C(x, y, \beta) + P(\beta)$$

- Two main types:

$$P(\beta) = \lambda \sum_{i=1}^{n} |\beta_i| \qquad\qquad P(\beta) = \lambda \sum_{i=1}^{n} \beta_i^{\,2}$$ [1]

LASSO Regression (L1 Regularization)

- **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
- First proposed by Robert Tibshirani in 1996
- Sum of the **absolute values** of the regression coefficients less than a fixed value
    - Turning certain coefficients to a value of zero, and eliminating them from the model

$$L(\beta) = \left[-\sum_{i=1}^{n} yi \log(\Pi_i) + (1 - yi)\log(1 - \Pi_i) \right] + \lambda \sum_{j=1}^{p} |\beta_i| \quad [2]$$

Penalty Term

- L1 Norm or L1 Penalty
- **Absolute value** of the magnitude of the coefficients, limiting the size of said coefficient

$$P(\beta) = \lambda \sum_{i=1}^{n} |\beta_i|$$ [2]

Key Points

- Lasso utilizes **shrinkage** in order to perform variable selection
  - Variables equal to **zero** are able to be eliminated from the model
- Produces sparse models with **few** coefficients
- Easily interpreted due to subset of predictors and **eliminated coefficients**
- Works best when number of predictors is consistently **less** than the number of observations
  - When λ is sufficiently **large**

Ridge Regression (L2 Regularization)

- First proposed by Hoerl and Kennard in 1970

- Introduces bias for less variance to reduce the mean square error

- Use when variables are **codependent** or **collinear**

$$L(\beta) = \left[-\sum_{i=1}^{n} yi \log(\Pi_i) + (1 - yi) \log(1 - \Pi_i) \right] + \lambda \sum_{j=1}^{p} \beta_j^2$$

Penalty Term

- L2 Norm or L2 Penalty

- **Sum** of the coefficients for each variable squared, multiplied by Lambda

- This penalty term seeks to ensure coefficient values stay small, as large coefficients will add too heavily to the cost function

$$P(\beta) = \lambda \sum_{j=1}^{p} \beta_j^2$$

Key points

- Achieve shrinkage when adding a term by **decreasing** the coefficients

- λ behaves similarly
  - As it **increases**, the coefficients are pushed closer **towards 0**

- Final coefficients **discern** what variables the model deems to be strongly indicative of the dependant variable

- All independent variables are used in ridge regression with coefficients **ranging from [-1,1]**

Elastic Net Regression

- **Combines** both Lasso and Ridge Regression methods
- Performs variable selection and regularization **simultaneously**
- Most appropriate where the dimensional data is **greater** than the number of samples
- Two regularization parameters, one for each penalty

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - x_i^T\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta}_j^{\ 2} + \alpha\sum_{j=1}^{m}|\hat{\beta}_j|\right)$$

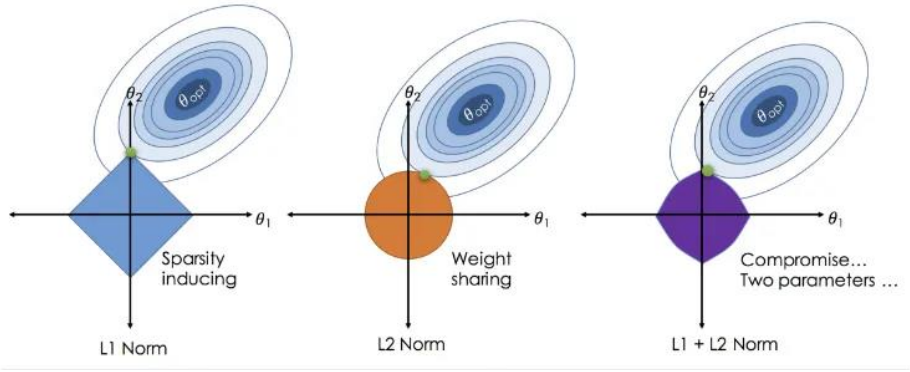
Penalty Term

- λ controls the **strength** of the penalty
  - If λ = 0, penalty terms equal and models are **without regularization**
  - If λ **increases**, so does penalty, and the **coefficients decrease**
  - If λ goes to **infinity**, coefficients shrink to nearly **0**

- Uses alpha as a hyperparameter to solve potential issues from either model
  - Lasso: poor performance when there is correlation
  - Ridge: cannot set values to 0

Geometric Comparison of Models

- **Lasso** performs variable selection by constraining the coefficient estimates to lie within a diamond shaped region
- **Ridge** shrinks the coefficient estimates towards 0 while maintaining relative sizes within a circular shaped region
- **Elastic** combines the sparsity-promoting and stabilizing effects within an intermediate shaped region



Advantages and Disadvantages of Each Model

Lasso Regression:

- Advantages:
  - Used for variable selection, handles a large number of predictor variables, performs well when the true model is sparse, and provides a solution that is easy to interpret
- Disadvantages:
  - May perform poorly when the true model is not sparse, may have difficulty in situations where there is high correlation between predictor variables

Ridge Regression:

- Advantages:
  - Handle a large number of predictor variables, performs well when the true model is not sparse, and used to reduce the impact of multicollinearity
- Disadvantages:
  - Does not perform variable selection, provides a solution that is less interpretable than Lasso, may not perform well when the number of predictor variables is larger than the sample size

Elastic Net Regression:

- Advantages:
  - Combines the advantages of both Lasso and Ridge, handles a large number of predictor variables, performs well in situations with high correlation between predictor variables
- Disadvantages:
  - May require more computational resources than Lasso or Ridge, provides a solution that is less interpretable than Lasso, may not perform well when the number of predictor variables is larger than the sample size

Finding λ via Cross-Validation

- Finding the **fittest λ** is of the utmost important in penalized regression

- λ **controls the weighting** of both penalties terms to the loss function

- How **CV** works
    - Divide the testing data in *k* groups "folds"
    - Train the model on (*k*-1) folds using the final k^{th} fold as a validation data using randomly selected values
    - **Calculate the Mean Squared Error** for each λ, and choose the one that **minimizes** the **MSE**

Dataset - Pima Indians Diabetes Dataset (1988)

Outcome:

1 denotes having diabetes

0 denotes not having diabetes

Sample size: **768**

Selection criteria:

- Female 21 or older
- Member of the Pima Nation

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1

LASSO Regression Code

Package Needed: library(glmnet)

```
{r}
# Set a random seed for reproducibility
set.seed(1)
# Perform cross-validation using L1 regularization and the binomial family for logistic regression
cv.lasso <- cv.glmnet(x, y, alpha = 1, family = "binomial")

# Fit a regularized logistic regression model using the lambda value that gives the smallest cross
-validation error
model <- glmnet(x, y, alpha =1, family = "binomial", lambda = cv.lasso$lambda.min)
```

Ridge Regression Code

```
{r}
# Set a random seed for reproducibility
set.seed(1)

# Perform cross-validation using L2 regularization and the binomial family for logistic regression
cv.ridge <- cv.glmnet(x, y, alpha = 0, family = "binomial")

# Fit a regularized logistic regression model using the lambda value that gives the smallest cross
-validation error
model <- glmnet(x, y, alpha =0, family = "binomial", lambda = cv.ridge$lambda.min)
```

Elastic Net Regression Code

```
{r}
# Set a random seed for reproducibility
set.seed(1)
# Perform cross-validation using L1 and L2 regularization and the binomial family
for logistic regression
cv.elnet <- cv.glmnet(x, y, alpha = 0.5, family = "binomial")

# Fit an elastic net logistic regression model using the lambda value that gives
the smallest cross-validation error
model <- glmnet(x, y, alpha = 0.5, family = "binomial", lambda = cv.elnet$lambda
.min)
```

Coefficients of our Models

LASSO Ridge

Elastic Net

	s0	s0	s0
(Intercept)	-7.5636864002	-6.8722710135	-7.4172873976
Pregnancies	0.1167031225	0.1017124608	0.1132654412
Glucose	0.0306511471	0.0268113310	0.0298613940
BloodPressure	-0.0130476042	-0.0112346586	-0.0125664773
SkinThickness	-0.0035148283	-0.0039135403	-0.0035265608
Insulin	-0.0009342941	-0.0006293995	-0.0008542789
BMI	0.0898511316	0.0793981585	0.0874545339
DiabetesPedigreeFunction	0.8040466706	0.7683238005	0.7915706356
Age	0.0120986058	0.0142895291	0.0124793018

TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

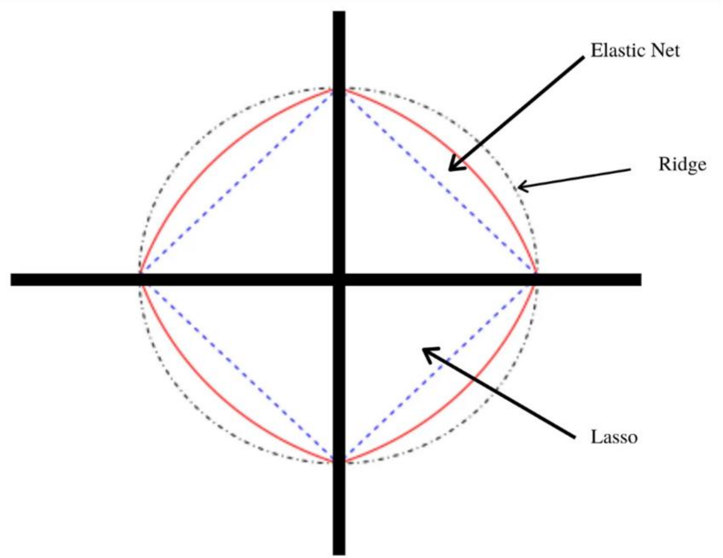
Results of the Models

	LASSO	**Ridge**	**Elastic Net**
Accuracy	0.812	0.812	0.818
Precision	0.5	0.467	0.5
Recall	0.833	0.875	0.857
F1-Score	0.625	0.609	0.632
Confusion Matrix	$\begin{array}{c\|cc} & 0 & 1 \\ \hline 0 & 126 & 30 \\ 1 & 6 & 30 \end{array}$	$\begin{array}{c\|cc} & 0 & 1 \\ \hline 0 & 128 & 32 \\ 1 & 4 & 28 \end{array}$	$\begin{array}{c\|cc} & 0 & 1 \\ \hline 0 & 127 & 30 \\ 1 & 5 & 30 \end{array}$

Precision = TP/(TP+FP)Recall = TP/(TP+FN)

F1-Score = (2*Precision*Recall)/(Precision+Recall)

Thank you. Questions?



References

[1] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Linear Model Selection and Regularization. In An introduction to statistical learning: With Applications in R (1st ed., pp. 203–264). Springer Texts in Statistics.

[2] Shofiyah, F., & Sofro, A. (2018). Split and conquer method in penalized logistic regression with Lasso (application on credit scoring data). Journal of Physics: Conference Series, 1108. https://doi.org/10.1088/1742-6596/1108/1/012107

[3] *Elastic net*. Corporate Finance Institute. (2022, December 28). Retrieved March 21, 2023, from https://corporatefinanceinstitute.com/resources/data-science/elastic-net/

[4] *Lasso*. Lasso and Elastic Net - MATLAB & Simulink. (n.d.). Retrieved March 21, 2023, from https://www.mathworks.com/help/stats/lasso-and-elastic-net.html

[5] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988), from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[6] Taboga, Marco (2021). "Ridge regression", Lectures on probability theory and mathematical statistics. Kindle Direct Publishing, from https://www.statlect.com/fundamentals-of-statistics/ridge-regression.

[7] Mohammed Alhamid (2020). "What is Cross Validation" , from https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75#:~:text=Cross%2DValidation%20has%20two%20main,in%20each%20fold%20or%20stratified.

[8] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1), 267–288. http://www.jstor.org/stable/2346178