Case 1

02582 Computational Data Analysis

s194244 & s194266

March 2023

1 Introduction and Data description

The main objective of this assignment is to build a predictive model of the real-valued response variable $\mathbf{y}$ by using the 100-dimensional feature matrix $\mathbf{X}$, which consists of two types of features; continuous and categorical. For this purpose, 100 observations have been provided $(y_n, \mathbf{x}_n)$, and an additional 1000 observations $\mathbf{x}_{new}$ without corresponding response values y_{new}. Hence, the objective is to compute the predictions $\hat{y}_{new}$ using the proposed predictive model and $\mathbf{x}_{new}$ while providing the estimated prediction error (EPE).

To achieve this objective, one has to propose a suitable predictive model given the available data $(y_n, \mathbf{x}_n)$. A possible solution is to rely on regression models, assuming an additive linear dependence between $\mathbf{y}$ and $\mathbf{X}$

$$\mathbf{y} = \boldsymbol{\beta}^T\mathbf{X} + \boldsymbol{\eta}, \quad \eta \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \tag{1}$$

and estimating the parameter vector $\boldsymbol{\beta}$ using ordinary least squares (OLS) [3]:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

Notice that the intercept β_0 is included in $\boldsymbol{\beta}$ and a column vector of ones has been added to $\mathbf{X}$ in equation 1.

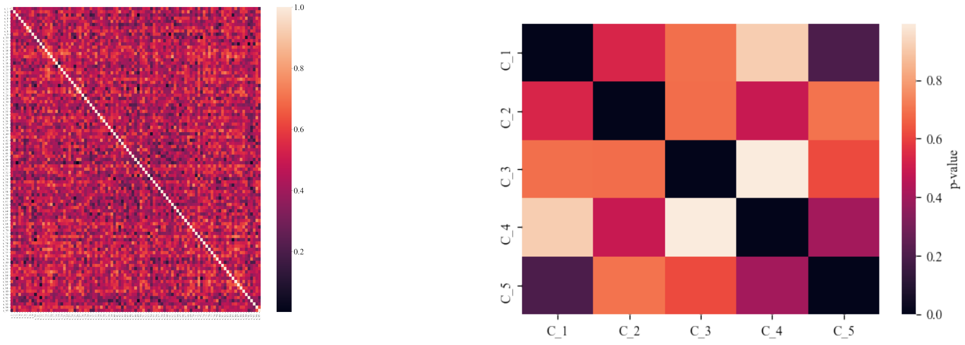
The relatively limited number of observations ($n = 100$) poses a challenging scenario because the number of features equals the number of observations ($p = n$), and if linear independence is assumed between the features, $p + 1$ observations is required to guarantee X^TX is invertible, which is necessary to obtain a unique solution for $\hat{\boldsymbol{\beta}}$ [3]. Section 2 explains our proposed approach to accommodate this issue. This challenge is further complicated by missing values, given that each row and column in $\mathbf{X}$ contain at least one missing value. Hence, excluding simple measures, e.g., dropping observations or features with missing values, this issue is further addressed in section 2.2.

If linear dependence between features is present, X^TX becomes singular, hence, non-invertible, and no unique solutions can be found [2]. However, linear dependence between features is not common unless the preprocessing has introduced redundant features [1]. A more common issue in practice is collinearity, which emerges if the columns of $\mathbf{X}$ possess strong-pairwise correlations. Collinearity introduces considerable variance in the coefficient estimation, which implies that the estimated values can differ significantly from the true values [2][1]. Figure 1a illustrates the estimated pairwise correlation between continuous features in $\mathbf{X}$ and shows the presence of multiple strong-pairwise correlations. Multiple pairwise χ^2 tests were used to detect potential collinearity between the categorical features. When multiple hypothesis testing is conducted, the significance level α has to be adjusted. Otherwise, the family-wise error rate (FWER) will increase. For instance, in this scenario, we conduct 10 independent tests with $\alpha = 0.05$, which yields the following FWER:

$$\begin{aligned} FWER &= 1 - (1 - \alpha)^M \quad (3a) \\ &= 1 - (1 - 0.05)^{10} \approx 0.40 \quad (3b) \end{aligned}$$

Hence, we have an approximately 40% probability of at least one false discovery instead of the intended 5%. To address this issue, we adopt the Bonferroni correction and re-scale α by the number of independent tests $\alpha_{BC} = \frac{\alpha}{M} = \frac{0.05}{10} = 0.005$. After re-scaling the significance level α, the FWER becomes $1 - (1 - 0.005)^{10} \approx 0.05$. Hence, using the re-scaled significance level, the probability of at least one false rejection of a true null hypothesis is approximately 5%.

In this context, the null hypothesis is the absence of a significant association between any pair of categorical or dichotomous variables. The results of the pairwise χ^2 test, summarised in figure 1b, yield no significant association at the significance level α_{BC}. Hence, the results do not indicate any major issues with collinearity among the categorical and dichotomous predictors.



(a) Pairwise correlation

(b) p-values of pairwise χ^2 tests

Figure 1: Collinearity of $\mathbf{X}$

2 Model and Method

As previously addressed, the data is ill-conditioned due to collinearity and potential rank deficiency. Regularisation is an often used technique to handle ill-conditioned data by imposing a bias to reduce the variance of coefficient estimates and make $\mathbf{X}^T\mathbf{X}$ non-singular [2][3]. We propose three different regularisation techniques, all of which penalize the norm of $\boldsymbol{\beta}$, to model the provided data: Ridge, Lasso, and ElasticNet regression. The three regression methods can be understood as minimizing equivalent loss functions but subject to different constraints as presented in equation 4.

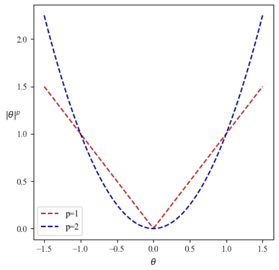


Figure 2: Regularisation behaviour

Group	*C1*	*C2*	*C3*	*C4*	*C5*
I	11	54	20	24	11
K	11	x	18	15	19
H	18	28	17	17	23
J	20	x	15	18	18
G	20	x	10	14	16
Missing	20	18	20	12	13

Table 1: Distribution of categorical variables. *C2* is a dichotomous variable

$$\hat{\boldsymbol{\beta}} = \underset{\beta_0, \boldsymbol{\beta}}{argmin}\{\mathbf{y} - \beta_0 - \mathbf{X}\boldsymbol{\beta}\} \quad s.t. \begin{cases} ||\boldsymbol{\beta}||_1 \leq t, & \text{if Lasso} \\ ||\boldsymbol{\beta}||_2 \leq t, & \text{if Ridge} \\ \alpha||\boldsymbol{\beta}||_1 + (1-\alpha)||\boldsymbol{\beta}||_2 \leq t, & \text{if ElasticNet} \end{cases} \tag{4}$$

Figure 2 illustrates the difference between l_1- and l_2-norm penalization of the coefficients. The most distinct difference is that the l_2-norm assigns less importance to small coefficient values, while the l_1-norm assigns them a higher contribution. Consequently, the constraint of Ridge regression will tend to shrink large coefficient values to minimize the loss efficiently, while Lasso will tend to set redundant parameters exactly to zero. Hence, Ridge is often seen as a shrinkage method, and Lasso as a continuous subset selection method [1].

ElasticNet is a compromise between l_1- and l_2-norm regularisation by introducing a weighted penalty balancing the emphasis on the two norms. The motivation for this method is to attain a single model, which can select features as Lasso and shrink correlated coefficients as Ridge regression [1].

All of the aforementioned regression methods are not equivariant to the input scale, which imply that $\mathbf{X}$ should be standardized prior to model fitting [1]. Furthermore, the response variable $\mathbf{y}$ is centered using the average $\bar{\mathbf{y}}$ estimated using the training observations, which entails that the intercept can be omitted, given that β_0 is estimated by $\bar{\mathbf{y}} = \frac{1}{N}\sum_{i=1}^{N} y_i$ [1]. When predicting on $\mathbf{X}_{new}$, the average $\bar{\mathbf{y}}$ prior to centering is added to all predictions $\hat{\mathbf{y}}_{new}$ because $\mathbf{y}_{new}$ is not observed, and therefore can not be centered using the average estimated on the training set. To avoid potential data leakage between training, validation and test splits these procedures are conducted on and using the training set.

2.1 Model Selection and Validation

Due to insufficient observations ($n = 100$), dividing the dataset into separate training, validation, and test sets for model selection and assessment is troublesome. Hence, we decided to rely on K-fold cross-validation for selecting the optimal model $\mathcal{M}_s^*$ w.r.t hyperparameter tuning. Furthermore, to mitigate potential issues with exceedingly optimistic *estimated prediction error* (EPE) caused by selecting the model with the lowest error when conducting hyperparameter tuning, we will use a two-step procedure, nested cross-validation, to ensure that one cross-validation procedure is used for selecting the best model $\mathcal{M}_s^*$ w.r.t hyperparameters, and another one for estimating the performance of $\mathcal{M}^*$ trained with equivalent hyperparameters as $\mathcal{M}_s^*$ [4]. This two-step procedure increases the number of computations. However, it ensures that the estimated prediction error is determined on an independent test set, which has not been used for model selection. To summarise, the inner cross-validation conducts the model selection w.r.t hyperparameter tuning. The outer cross-validation performs model assessment using the best-performing model determined by the inner cross-validation. As a trade-off between bias and variance, we have decided to use $K_{outer} = K_{inner} = 10$ folds, which implies 90 and 10 observations for training and testing in the outer fold, and 81 and 9 observations in the inner fold for training and validation. The splits in both cross-validations are seeded to ensure the proposed models are trained and evaluated on the same subsets of data.

Furthermore, the observations should be independent to avoid information leakage from the test and validation set to the training set. Hence, we permute the data before any splitting of the observations to ensure that there is no predefined order of the data that might cause leakage. We standardise the training data separately for each fold using the procedure explained in section 2. Lastly, we impute missing values separately for each training set as explained in section 2.2.

2.2 Missing Values and Factor Handling

We propose two different methods of handling missing values. The simple approach imputes missing continuous values by using the mean value estimated on the training set and categorical missing values by using the mode / most frequent category in the training set. Imputing with the mean value is common, given that it preserves the estimated mean of a feature after the imputation. However, if the distribution of the feature is skewed, it might be sub-optimal compared to imputing with the median. Using the most frequent category can also introduce class imbalance and lead to potential drawbacks, e.g. if the missing values carry information. The second approach uses k-nearest neighbour (kNN) imputation on the continuous and categorical features. For the continuous features, a missing value is imputed by the euclidean distance weighted average using the k-nearest neighbours in the training set. For imputing missing categorical values, a kNN classifier is trained using the training set with the category as the response variable and the continuous features as input. Subsequently, the kNN classifier predicts the missing categorical values. The kNN imputations assume the presence of associations between predictors, which for the continuous features seems reasonable based on figure 1a. For the categorical kNN imputations, we assume a relationship between continuous and categorical features, which we have not tested, and might be a limitation. Further, the hyperparameter k in both kNNs has been predetermined as $k = 5$, which is a limitation given that it should have been optimised equivalently as other hyperparameters.

As seen from table 1, we assume that there are one binary variable and four categorical ones. As the proposed models require numeric inputs [3], we one-hot encoded the four categorical features creating c features for each of them. Here, $c = 5$ is the number of assumed classes in the four features. The binary variable was converted to numeric values (0, 1). Consequently, the factor handling also eliminates the use of standard OLS even without splitting the provided data because $p > n$.

3 Results

Table 2 illustrates the results of the two-step nested cross-validation procedure for each model with the two proposed imputation methods; mean and most frequent (MF), k-Nearest Neighbour (kNN). The table reports, for each outer fold, the best-performing hyperparameter, e.g. λ^*, selected by the inner cross-validation and the corresponding test error. The estimated prediction error for each of the proposed models is reported in table 3 and is determined by $EPE = \frac{1}{K_{outer}} \sum_{i=1}^{K_{outer}} E_i^{test}$.

	HP		*Test error*	
ith Fold	λ^*_{MF}	λ^*_{kNN}	$E^{test}_{i,MF}$	$E^{test}_{i,kNN}$
1	0.20	0.13	27.41	24.4
2	0.53	0.17	41.80	37.08
3	0.31	0.23	35.30	30.72
4	0.40	0.31	29.71	26.26
5	0.53	0.35	30.61	24.99
6	0.26	0.40	22.58	17.52
7	0.61	0.35	28.89	26.44
8	0.53	0.15	26.98	29.74
9	0.35	0.15	18.75	20.30
10	0.40	0.23	27.57	21.63

	HP		*Test error*	
ith Fold	λ^*_{MF}	λ^*_{kNN}	$E^{test}_{i,MF}$	$E^{test}_{i,kNN}$
1	0.19	0.26	19.88	12.9
2	0.34	0.26	39.64	39.62
3	0.45	0.11	34.78	29.33
4	0.26	0.26	26.79	20.33
5	0.34	0.26	21.77	17.61
6	0.45	0.34	18.46	15.76
7	0.34	0.26	23.00	19.66
8	0.34	0.19	20.77	28.74
9	0.34	0.26	21.80	13.78
10	0.34	0.26	29.83	21.10

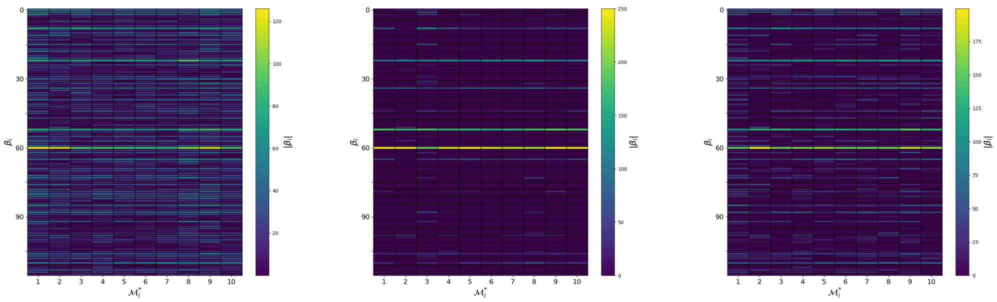
	HP				*Test error*	
ith Fold	α^*_{MF}	α^*_{kNN}	λ^*_{MF}	λ^*_{kNN}	$E^{test}_{i,MF}$	$E^{test}_{i,kNN}$
1	0.99	0.97	0.15	0.04	21.14	20.57
2	0.99	0.99	0.15	0.06	41.75	38.34
3	0.96	0.99	0.06	0.08	35.40	30.05
4	0.99	0.99	0.15	0.11	29.49	26.21
5	0.99	0.99	0.19	0.11	27.79	22.18
6	0.93	0.99	0.05	0.11	19.23	16.74
7	0.99	0.99	0.26	0.08	25.99	21.03
8	0.99	0.99	0.19	0.11	26.29	29.52
9	0.99	0.99	0.15	0.06	19.68	15.14
10	0.99	0.99	0.11	0.11	29.02	22.17

Table 2: Nested cross-validation

	Ridge		*Lasso*		*ElasticNet*	
	Ridge$_{\mathrm{MF}}$	Ridge$_{\mathrm{KNN}}$	Lasso$_{\mathrm{MF}}$	Lasso$_{\mathrm{KNN}}$	ElasticNet$_{\mathrm{MF}}$	ElasticNet$_{\mathrm{KNN}}$
EPE	28.96 ± 4.01	25.91 ± 3.57	25.67 ± 4.44	$\mathbf{21.88 \pm 5.28}$	27.58 ± 4.45	$\underline{24.25 \pm 4.39}$

Table 3: Model comparison w.r.t estimated prediction error (EPE) ($\pm 2SE$)

The results in table 2 illustrate that the hyperparameters selected by the inner-cross validation are more stable for Lasso and ElasticNet compared with Ridge regression. Thus, it appears that the optimization procedure for ridge regression is unstable, given that different models are selected when different subsets of the data are used. Furthermore, the selected hyperparameter α^* in ElasticNet consistently approaches 1 for each fold regardless of the imputation method, which entails that the selected model heavily prioritizes the l_1-norm regularisation over l_2, meaning that the selected ElasticNet models resemble the Lasso. Lastly, the imputation method kNN generally outperforms the other method across the different folds for each model.



(a) Ridge regression (b) Lasso regression (c) ElasticNet regression

Figure 3: Sparsity coefficient pattern of $\mathcal{M}_i^*$

The EPEs presented in table 3 show that Lasso with kNN imputation obtained the lowest EPE with slightly more uncertainty than the other models. Figure 3 illustrates the absolute value of the estimated coefficients for each of the selected models $\mathcal{M}_i^*$. By inspecting the sparsity patterns of the three proposed models, the difference between l_1- and l_2-norm regularisation becomes apparent. The selected ridge regression models possess multiple coefficient estimates with small absolute values, while the selected Lasso regression models tend to have few non-zero coefficients across all folds, which is aligned with the behaviour illustrated in figure 2. Furthermore, the sparsity pattern of the selected ElasticNet models resembles the pattern of the Lasso models, however, with more non-zero coefficients and smaller absolute coefficient values, which aligns with the motivation of the combined penalisation. In general, the lasso regression provides the most parsimonious model of the three considered as expected.

All EPEs overlap in terms of their uncertainty which makes it difficult to make a confident decision based on EPE alone. Therefore, we select the best model in terms of all the previously mentioned characteristics. Based on all of these points, the chosen final model is Lasso with a kNN imputing. Hence, the selected model is fitted on all of the provided data $(\mathbf{y}, \mathbf{X})$ using the mode of the selected hyperparameters ($\lambda = 0.26$) seen in table 2. This model is used to predict $\hat{\mathbf{y}}_{new}$ using $\mathbf{X}_{new}$.

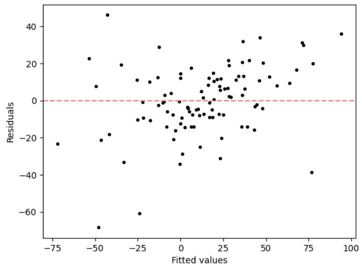


Figure 4: Residual plot (Lasso - kNN imputation)

The residuals vs fitted values shown in figure 4 originate from the final model trained on $\mathbf{y}$ and $\mathbf{X}$. The figure indicates that there may be a minor issue with heteroscedastic behaviour due to the slightly non-constant variance, which might indicate that the final model may not fully explain the variance of $\mathbf{y}$, which is a potential drawback of choosing the most parsimonious model. Furthermore, the residuals vs. fitted do not appear to illustrate any non-linear trends, which indicates that the proposed linear models are reasonable. Finally, there seem to be potentially one or two outliers. However, due to the small number of observations as well as our limited knowledge of the data sources, outlier removal does not seem justified.

4 Code

Code is provided at the following GitHub repository: `https://github.com/Ne0-1/02582_computational_data_analysis`

References

[1] Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction.* Springer, 2017.

[2] Henrik Madsen and Poul Thyregod. *Introduction to general and generalized linear models.* CRC Press, 2011.

[3] Sergios Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective.* Elsevier Ltd, 2015.

[4] Morten Mørup Tue Herlau, Mikkel N. Schmidt. *Introduction to Machine Learning and Data Mining - 02450.* 2020.