Generalized Linear Models

Biljana Vitanova
MLDS1 2024/25 , FRI, UL
bv7063@student.uni-lj.si

I. Introduction

The goal of this project is to implement and compare Multinomial and Ordinal Logistic Regression using a synthetic data generating process. We then apply and interpret the multinomial model on a basketball dataset, and explore common GLM diagnostics in the context of linear regression model.

II. Model Implementation

A. Methodology

We implement two models Multinomial Logistic Regression and Ordinal Logistic Regression. Multinomial Logistic Regression extends binary logistic regression to more than two classes. With n classes and m predictors, the model learns $(m+1) \times (n-1)$ parameters. One class is fixed to prevent identifiability issues, and we add one extra predictor value for the intercept.

The model uses the softmax function as the link function. For a sample $\mathbf{x}^{(i)}$, the probability of class c is:

$$P(y^{(i)} = c \mid \mathbf{x}^{(i)}) = \frac{\exp(\mathbf{x}^{(i)\top}\beta_c)}{\sum_{j=0}^{n-1} \exp(\mathbf{x}^{(i)\top}\beta_j)}$$

The model is trained to give high probability to the correct class labels, or in other words, to maximize the likelihood. The likelihood over all samples is:

$$L(\beta) = \prod_{i=1}^{N} \prod_{c=0}^{n-1} \left[P(y^{(i)} = c \mid \mathbf{x}^{(i)}) \right]^{\mathbb{I}[y^{(i)}=c]}$$

To simplify the expression, we take the log of the likelihood and then minimize the negative log-likelihood, which leads to the same result.

The other method, Ordinal Logistic Regression is used when the class labels are ordered. With n classes and m predictors, the model learns $m+1$ parameters for the weight vector (including the intercept) and $n-2$ thresholds. One threshold is fixed to 0 to prevent identifiability issues.

The model uses the logistic sigmoid as the link function. For each sample $\mathbf{x}^{(i)}$, we compute a score $\mathbf{x}^{(i)\top}\beta$, and the class probabilities are:

$$P(y^{(i)} = c \mid \mathbf{x}^{(i)}) = \begin{cases} \sigma(\theta_1 - \mathbf{x}^{(i)\top}\beta) & \text{if } c = 0 \\ 1 - \sigma(\theta_{n-1} - \mathbf{x}^{(i)\top}\beta) & \text{if } c = n-1 \\ \sigma(\theta_{c+1} - \mathbf{x}^{(i)\top}\beta) - \sigma(\theta_c - \mathbf{x}^{(i)\top}\beta) & \text{otherwise} \end{cases}$$

The likelihood definition and the training process (minimizing the negative log-likelihood) is similar as in the multinomial case.

B. Results and Discussion

To test the correctness of our implementations, we generated synthetic data with 300 samples and three classes. We compared our custom implementation of multinomial logistic regression with the one from `sklearn`. The results were similar, with a difference in accuracy of less than 0.1. For ordinal logistic regression, we created a simple dataset of 15 values, with 5 samples for each of the classes 0, 1, and 2. The model was able to predict the correct class for each sample in the test set.

III. Application of the Multinomial Regression

A. Methodology

Next, we aim to understand how the predictors relate to the target variable `ShotType`, or more simply, we interpret the estimated β coefficients of the model. First, we transform the features. Categorical features are one-hot encoded. To avoid multicollinearity, we drop one category from each set. After encoding, we check for multicollinearity between the features using the Variance Inflation Factor (VIF). High VIF values suggest that a feature is strongly correlated with other features. We remove the feature with the highest VIF and recalculate VIFs again. This is done heuristically until all values are in an acceptable range. To estimate the uncertainty of the coefficients, we apply bootstrapping. We repeatedly resample the dataset with replacement, fit the model on each sample, and compute the coefficient estimates. After that we calculate the 95% bootstrap confidence intervals for each coefficient.

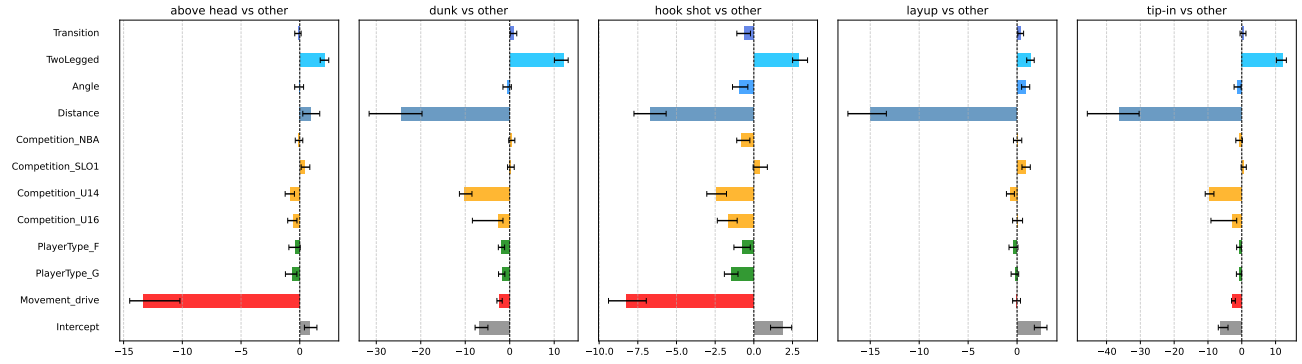


Fig. 2: β coefficients of features for each `ShotType` class, with reference class `other`.

B. Results and Discussion

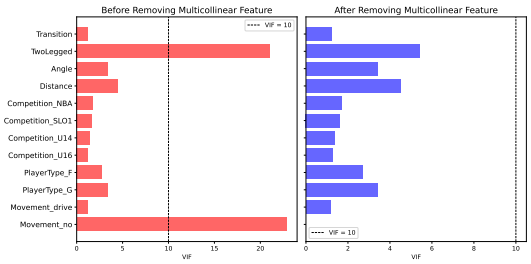


Fig. 1: VIF values for the full training set (left), and after removing highly correlated features (right).

From Figure 1, we can see that in the full training set, there are two features, `TwoLegged` and `MovementNo`, which have VIF values greater than 20, which suggests high correlation with the other features. After removing `MovementNo`, the VIF value for `TwoLegged` decreases to approximately 5. Since all remaining features have VIFs less than or equal to 5, we include them in the further step.

From Figure 2, we focus on the features with the largest absolute β values, particularly *Movement_Drive* and *Distance*. For the class *above head*, the coefficient for *Movement_Drive* is strongly negative (approximately -12), meaning that as this feature increases, the log-odds of a shot being classified as *above head* (instead of the *other* class) decrease. This is intuitive *above head* shots are typically taken in static situations, and are less likely to occur during fast drives. For the remaining four classes *dunk*, *hook shot*, *layup*, and *tip-in* the coefficient for *Distance* is strongly negative. This means that as the distance increases, these types of shots become less likely to occur compared to the *other* category, which is expected since they typically happen close to the basket. Another feature that has positive coefficients across all classes is *TwoLegged*. This means that performing a shot with both legs is more common across all shot types compared to the *other* category.

IV. Application of the Ordinal Regression

A. Methodology

We define a data-generating process with one feature sampled from a normal distribution $x \sim \mathcal{N}(0,1)$. We multiply the values by a constant β and add noise $\epsilon \sim \mathcal{N}(0,1)$. Then we convert the result into 5 class labels using fixed thresholds.

B. Results and Discussion

Sample Size	Model	Log-Loss
100	Multinomial Regression	0.8495 ± 0.5052
	Ordinal Regression	0.4510 ± 0.6707
1000	Multinomial Regression	0.0990 ± 0.0106
	Ordinal Regression	0.0924 ± 0.0062

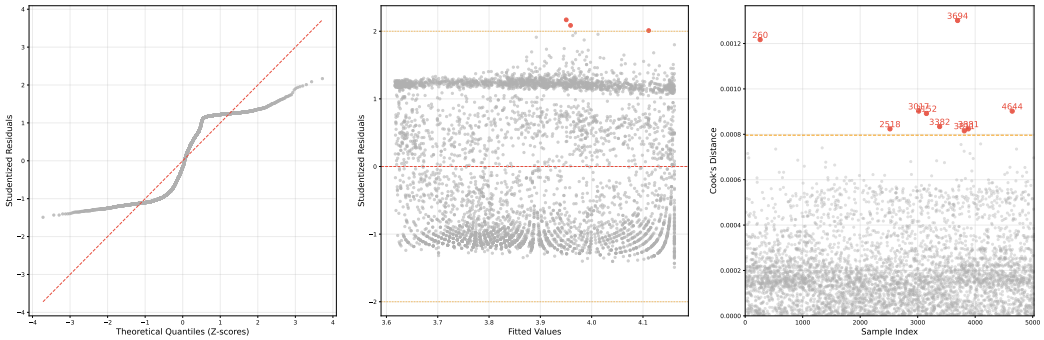
TABLE I: Log-loss comparison between Multinomial and Ordinal Logistic Regression using a custom DGP.

From Table I, we can see that with a smaller training sample, the ordinal regression outperforms the multinomial regression. As we increase the sample size, both models perform similarly.

V. GLM Diagnostics

A. Methodology

In the next section, we fit a linear regression model to predict shot distance based on the angle feature. After fitting the model, we check how well it performs using three tools: a Q–Q plot, a residuals vs fitted values plot, and Cook's Distance. First, we choose the right type of residuals. Since linear regression assumes that the residuals are normally distributed, we use studentized residuals. These are better than raw residuals because they account for leverage, how much influence each data point has, and make the residuals more comparable across all points. We use these studentized residuals for the Q–Q plot, which helps check whether the residuals follow a normal distribution (as expected), and for the residuals vs fitted plot, which helps



(a) Normal Q-Q Plot (b) Residuals vs Fitted Values (c) Cook's distance

Fig. 3: Diagnostics of the linear regression model predicting distance from angle in the basketball dataset.

reveal problems like non-linearity or non-constant variance. Finally, we calculate Cook's Distance to see if any data point has too much influence on the model's predictions.

B. Results and Discussion

From Figure 3a, we can see that the fitted residuals do not closely follow the theoretical quantiles (Z-scores). The curve forms an S-shape, which clearly suggests that the residuals are not normally distributed, violating one of the key assumptions of linear regression.

In Figure 3b, it's difficult to say for sure whether the variance is constant across the data. The residuals mostly fall in the range of approximately [–1.2 to 1.2], and there's no obvious "V" shape that would indicate changing variance. However, there are some visible patterns, especially in the lower residual range (around –1.2), which could suggest slight structure in the residuals, thus not constant variance.

In Figure 3c, all Cook's Distance values are well below the common threshold of 1, and even below 0.5. We highlighted points above the 4/n threshold as potentially influential. However, even the largest Cook's Distance is around 0.0012, so it's unlikely that any single point is strongly influencing the model.

VI. Conclusion

In this project, we implemented multinomial and ordinal logistic regression models and compared their performance on a synthetic dataset. The ordinal model performed better when the target variable had a clear ordering and the sample size was small, while performance became similar as the dataset grew. We also applied multinomial regression to a real-world basketball dataset and interpreted the model coefficients. Finally, we used GLM diagnostic tools and observed that the assumptions of linear regression were clearly violated.