Logistic regression

Matej Miočić, 63180206, mm9520@student.uni-lj.si

I. Introduction

We were tasked with implementation of multinomial logistic regression and ordinal logistic regression. Using (log-)likelihood of each model, we can train the model to make predictions for new observations. Since these 2 methods do not have a closed solution, we use Limited-memory BFGS optimization algorithm. We also try to interpret the coefficients of the categorical logistic regression on a dataset of basketball shot types. In the end we provide a data generating process where ordinal logistic regression has a better log score than multinomial logistic regression.

II. Models

A. Multinomial logistic regression

Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems. As in logistic regression, we assume a linear relationship between a set of weights and features. Since we have more than two classes, we represent weights in a matrix where each weight corresponds to each class and feature. Because we want our weights to be more consistent, we select a reference category. We decided to select the last category to be the reference category. This means that the last column of weighs corresponding to the reference category will be all zeros. We assume the target variable follows a categorical distribution. We define our model as (1):

$$y_i|\beta, x_i \sim Categorical(softmax(u_{1i}, u_{2i}, ..., u_{(m-1)i}, 0)), \quad (1)$$

where $u_{ji} = \beta_j^T x_i$, softmax is our link function and m is the number of categories.

B. Ordinal logistic regression

Ordinal logistic regression is a regression model for ordinal dependent variables. Since we assume ordinality in our data, our model is simplified making it more robust and easier to interpret. If we have m categories, we need m-1 breaks to split each of them. Just like before we set a "reference break" to 0. We decide to choose the first break as the reference break. We assume standard logistic noise which leads us to ordinal logistic regression. We define our model as (2):

$$y_i|t, \beta, x_i \sim Categorical(p_i), \quad (2)$$

where $p_i(j) = F(t_j - u_i) - F(t_{j-1} - u_i), j = 1..m$ and F is the CDF of the standard logistic distribution and $u_i = \beta^T x_i$.

C. Implication of multinomial logistic regression

We use multinomial logistic regression on a given dataset of basketball shots in real-world basketball games. We try to predict a type of basketball shot based on numerous features, such as **leg position** (one-legged and two-legged), **movement** (dribble or cut, drive and no movement), **player type** (guard (G), forward(F), and center (C)), **distance**, **angle**, **transition** and **level** (EURO, NBA, SLO1, U16 and U14). We use categorical regression to provide insights into the relationship between shot type and the other variables.

1) Dataset preparation: To remove ordinality assumption from multi-class features we use one-hot encoding. But with that we introduce full colinearity. This is why we remove one class for each feature. We decided to remove: SLO1 from **level**, guard (G) from **player type** and no movement from **movement**. To speed up convergence we standardized our continous features **distance** and **angle**. As we decided in our model introduction we have to choose a reference category. We chose **other** as our reference category. We also decide to include the intercept in our model.

2) Coefficients interpretation: Our main task is to provide insights into the relationship between shot type and the other variables and interpret the coefficients. Since coefficients contain uncertainty we decided to construct bootstrapped 95% symmetric percentile confidence intervals. We used 100 bootstrapped datasets and provide intervals of 5th and 95th percentile. We used the whole training set for bootstraping.

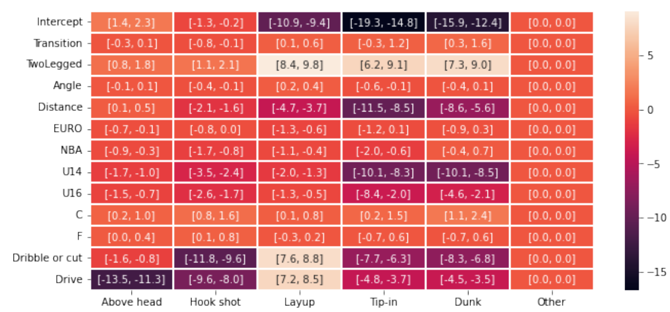

Figure 1. A heatmap of coefficients with bootstrapped 95% symmetric percentile confidence intervals. We see our shot types on the x axis, and features on y-axis. Coefficients for **other** category are set to 0, since this is our reference category.

We will try to make sense of coefficients in Figure 1 for each provided feature (before one-hot encoding):

1) The **intercept** tells us how our model predicts a class if all features are set to 0. We see that the most likely predicted shot types are **above head** and **other** if all features are set around 0.
2) **Transition** tells us if the attack was made when while the defending team was still in transition from attacking to defending. In that position dunking makes the most sense, since it guarantees points. It can be seen from the heatmap that dunking while in transition has the highest coefficient out of all transition coefficients, which proves our hypothesis.
3) **Two-legged** feature classifies whether a shot was made on one leg or both legs. We see that layup, tip-in and dunk are strongly correlated with using both legs, meanwhile surprisingly above head shots do not contribute as much.
4) **Angle** does not define a type shot as good as other features, this is why all coefficients are relatively low and close to each other.
5) **Distance** on the other hand defines a shot type really well. We see that layup, tip-in and dunk coefficients are more on the negative side than other possibilites. Which

makes sense, since you have to be close to the hoop to perform those types of shots.

6) Next we compare 5 **different levels** of competition (we must not forget the removed SLO1 feature). Coefficients tell us that in senior (NBA, EURO and SLO1) competitions we see a lot more tip-ins and dunking than with youth (U16 and U14). Which makes sense, since more experienced, taller players are more likely to perform those types of shots.
7) **Player type** is not as significant as in football, since we have fewer players. We do observe higher coefficients in close types of shots (tip-in, dunk) in center (C) players, but not in forward (F), which would be expected.
8) **Movement** feature is a bit harder to understand. Most of types of shots (above head, hook shot, tip-in and dunk) have negative coefficients which means that no movement has the most positive influence on predicting with this feature. However we have to remember these are all compared to **other** types of shots, besides **other**. Which means dunking is less likely to be performed with drive than **other**, but more likely than all other types except layup. Which makes sense.

D. Data generator

For the last task we had to implement a data generating process where ordinal logistic regression has a better log score than multinomial logistic regression. We explained in the introduction that with ordinal logistic regression we assume ordinality which in turn makes our model simpler. That means that the model has fewer parameters. As long as our assumptions hold (target variable has ordinality) and our training data set is sufficiently small, ordinal logistic regression should perform better. To demonstrate this we implemented a data generator which creates a dataset with 2 features. First feature contains numbers uniformly generated from -10 to 10. Similarly second feature contains numbers uniformly generated from -1 to 1. We generate 2 features to make our data complex enough so that some training is still needed to be done. We decide for our target variable to contain 3 classes (0 to 2). 0 class is set when feature 1 is smaller than -5 and feature 2 smaller than -0.5. Similarly we set our target variable to 2 if feature 1 is larger than 5 and feature larget than 0.5. The rest of the target variables are class 1. When we run this through our models with training set size 10 and test set size 1000, we compute log loss and obtain the following results (see Table I).

Table I
Log scores of multinomial and ordinal logistic regression on training set of size 10 and test set of size 1000 generated by our data generator.

	Log score (train)	Log score (test)
Multinomial	7.61e-07	12.18
Ordinal	0.56	0.84

We see that log score for multinormial is better on the training set, but we see the opposite when comparing log score for test dataset.

III. Conclusion

We learned how to implement multinomial and ordinal logistic regression and understand its process and coefficients. We have shown a practical explanation of coefficients on a dataset of basketball shot types. Lastly we have shown that multinomial logistic regression can be used for a dataset with ordinality, but if we have a small training dataset, ordinal logistic regression might outperform it.