Report on Ordinal Regression in fulfilment of requirements for the course Statistical Machine Learning

Christina Kapatsori, Pranav Krishna

Keywords: Ordinal Regression, Ordered Classes, POM, Partial POM, Random Forest, OrdinalNet

1. Introduction

Ordinal Regression is a model which arises in contexts where the response variable belongs to one of several ordered categories. For example, the progression of some disease, spiciness of curry or the levels of pain. The important thing in this model is the fact that the class labels are imbued with order information and the misclassification costs are not the same for each class. This is a middle ground between Regression estimation and Pattern recognition as mentioned in (5).

The paper assigned to us, (1) is a survey paper that develops a taxonomy for the methods of Ordinal Regression and does experiments with a few of the methods discussed. Notably, the paper does not consider the non-parallel case for POM and forest-based classifiers in detail.

In this report, we provide the theoretical underpinnings and experiments of the non-parallel case in section 2 and the results of using isotonic regression to calibrate probabilities from various setups in 3.

To state the problem concretely, the goal is predicting the output label $\mathbf{y}$, given the data $\mathbf{X}$, where $\mathbf{y} \in \mathcal{Y} = \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$ and $\mathbf{X} \in \mathcal{X} \subseteq \mathbf{R}^k$. The crucial piece of information is the order information of the class labels, via $\{\mathcal{L}_1 \prec \mathcal{L}_2, \ldots \prec \mathcal{L}_n\}$. Thus, the misclassification costs are not the same for all errors.

2. Approach 1: POM

In this section, we describe the structure of the POM and its generalisations. We conclude with some experimental results about it.

2.1. Developing the Proportional Odds Model

For simplicity let us consider a data-set where the outcome has 3 possible categories. If we denote the CDF of a standard logistic distribution as F_Z the POM model for $m \in \{1, 2\}$ is:

$$\mathbb{P}(Y \leq m | X = x) = logit^{-1}(\alpha_m + x^T\beta) = F_Z(\alpha_m + x^T\beta) \tag{2.1}$$

We can fit a set of thresholds which divide the real line into segments (as many as the categories) and we can assume the existence of a real-valued latent variable y^*, determined by: $y^* = x^T\beta + \epsilon$, where ϵ is the random error component with zero expectation distributed according to F_ϵ. We notice that $x^T\beta$ shifts the probability towards either higher or lower categories. We would like to construct a model with more flexibility in terms of how the probability can be shifted as mentioned in (2). To recap, we have that $y = \begin{cases} 1, & y^* \leq \theta_1 \\ ... & ... \\ k, & \theta_{k-2} \leq y^* \leq \theta_{k-1} \end{cases}$.

2.2. Generalizing the Proportional Odds Model

We introduce the elementwise link multinomial-ordinal (ELMO) class of models. Each model in this class is defined in terms of a family(e.g., cumulative probability) and a link function (e.g. logit).

An important feature of the ELMO class is that each model has a form that is appropriate for ordinal response data, as well as a more flexible form that can be applied to both ordinal or unordered categorical responses. We will refer to the first as the parallel form and the second as the non parallel form. For the parallel form, the linear predictors of a given observation only differ by the intercept values - the other coefficients are the same (also knows as the parallel assumption). The nonparallel form allows all for the coefficients to vary. An example from the ELMO class is exactly the proportional odds model, which is a parallel model that has a non-parallel counterpart, the partial proportional odds model (3).

Finally, we propose an elastic net penalty that applies to both the parallel and nonparallel forms. It can also be used to shrink the nonparallel model toward its parallel counterpart. This can be useful in a situation where one would like to fit an ordinal model but relax the parallelism assumption. This can be achieved by over-parameterizing the nonparallel model to include both the nonparallel and parallel coefficients. We call this alternate parameterization the semi-parallel model. Although the model itself is not identifiable under this parameterization, the penalized likelihood has a unique optimum (6).

2.3. Specifying the models for $\textbf{\textit{logit}}(\mathbb{P}(Y \leq m|X = x)) = ...$

The parallel model gives us that $= \alpha_m + x^T\beta$
We note that a change in $x^T\beta$ will shift all cumulative probabilities in the same direction as shown in 4. The non-parallel model gives us that $= \alpha_m + x^T\beta_m$, placing no restriction on β_m. Thus, it does not force the cumulative class probabilities to “shift together” in any way as shown in 5.

We note that a regularized parallel model could outperform a regularized nonparallel model, but how “parallel” does the data need to be in order to make the parallel model the better choice?
With these remarks stated, the semi-parallel model is proposed which is ordinal in nature and allows deviation from the parallelism assumption.
We define the semi-parallel model as:$= \alpha_m + x^T\beta + x^T\beta_m$
This model along with the Elastic Net Penalty allows for generalized regularization with even more flexibility on shifting x^TB as shown in 6.

2.4. Preventing Over-fitting : The Elastic-Net Penalty

The elastic net penalty is a weighted average between the lasso and ridge regression penalties,. It shares the lasso property of shrinking some coefficients to zero exactly. The weighting parameter, typically denoted by α, must be selected or tuned on the data set. The degree of penalization is controlled by another tuning parameter, typically denoted by λ. Typical practice is to fit the penalized model for a sequence of λ values and use a tuning procedure to select the best value. We use cross-validation. We refer the reader to the appendix at 5.1 for the expressions.

2.5. Experimental Results

The data-set Hccframe containes Methylation levels of 46 CpG sites from 56 human subjects. The measurements come from liver tissue samples where 20 subjects have a normal liver, 16 have cirrhosis (disease) and 20 have hepatocellular carcinoma (severe disease).

We used the package OrdinalNet, choosing $\alpha = 0.5$ for the Elastic Net Penalty, so that both Lasso and Ridge Penalty are applied and performed 5-fold Cross Validation to tune the hyperparameter λ within each training sample, and the value with the best average out-of-sample log-likelihood was selected. All 3 models (parallel, non-parallel, semi-parallel) were implemented this way and finally compared on the basis of misclassification and log-likelihood to determine the one that performed best. We note that in this example and in all examples mentioned in the Appendix the semi-parallel model had the best performance.

The Misclassification rate of parallel-model was 0.07 and the log-likelihood was -2.3. The Misclassification rate of semi-parallel model was 0.07 and the log likelihood was -2.1. Finally, the Misclassification rate of non-parallel model was 0.53 while the log-likelihood was -11.3.

We see that the semi-parallel model performs the best, even though the non-parallel and the semi-parallel models are the same analytically, but their performance is vastly different. The parallel model performs in a satisfactory way as well (see 3) which we may have expected, but the non-parallel is much worse since within the first few λ values we have non-monotone cumulative probabilities (which is a constraint in the non parallel model).

3. Approach 2: Ordinal regression using Classifiers

In this section, we describe how any binary classifier can be used for ordinal regression after partitioning the data and using isotonic regression to calibrate the output probabilities. Finally, we conclude with some experimental results. We would like to point out that classification approached can use ordinal data as predictors in a straightforward way while the POM approach relies on embedding such predictors in the real line.

3.1. Separating the data

While we observed in 1 that treating the entire problem as one classification problem is incorrect, we can overcome this issue in a simple manner. The essential insight is to partition the data correctly. Thus, instead of the "One vs one" or "One vs All" paradigms of partitioning the data, we partition it differently. Suppose we had 5 classes in the data. Then, we can split the data in one of the following ways as mentioned in Figure 1, Now, for each classifier, we consider the output in terms of probabilities and instead of thinking of them as conditional probabilities, we say that they are the unconditional probabilities wherein the classifier was trained using only the "difficult" observations. Because these probabilities are not necessarily non-decreasing, we calibrate them using isotonic regression.

$$\begin{array}{cccc}
\textit{OrderedPartitions} & \textit{OneVsNext} & \textit{OneVsFollowers} & \textit{OneVsPrevious} \\
\begin{pmatrix} -,-,-,- \\ +,-,-,- \\ +,+,-,- \\ +,+,+,- \\ +,+,+,+ \end{pmatrix} &
\begin{pmatrix} -,\ ,\ ,\ \\ +,-,\ ,\ \\ \ ,+,-,\ \\ \ ,\ ,+,- \\ \ ,\ ,\ ,+ \end{pmatrix} &
\begin{pmatrix} -,\ ,\ ,\ \\ +,-,\ ,\ \\ +,+,-,\ \\ +,+,+,- \\ +,+,+,+ \end{pmatrix} &
\begin{pmatrix} +,+,+,+ \\ +,+,+,- \\ +,+,-,\ \\ +,-,\ ,\ \\ -,\ ,\ ,\ \end{pmatrix}
\end{array}$$

Figure 1: Ways of partitioning the data before using classifiers. Here, a + represents the positive class, the − represents the negative class and the classes not considered are left blank and each column represents one classifier

3.2. Calibration

Isotonic regression is used to fit a monotonic piece-wise continuous constant function as closely to the data as possible. Thus, given a set of points $\{(x_1, y_1), \ldots (x_n, y_n)\}$, we seek real numbers $\{p_i\}_{i=1}^{n}$ such that $\Sigma_{i=1}^{n}(y_i - p_i)^2$ is minimised subject to $p_i \leqslant p_{i+1} \forall 1 \leqslant n-1$.

We remark that the loss function is the same as linear regression. We use this to calibrate the probabilities from all classifiers of a chosen partition of the data to give the final output, either as a vector of probabilities or the class. In our case, e.g., we see that the output of a classifier using the "OnevsNext" partition may not be monotonic. Thus, we sum the appropriate probabilities and use isotonic regression to calibrate them so that they are non-decreasing. We can use the output after calibration to make a prediction about the samples.

3.3. Experimental Results

For evaluating this method, we used the randomForest function available in package of the same name available in R as our binary Classifier. We used three data-sets for all the partitions mentioned in 1 and report the results in terms of the Confusion matrix for one of them.

For the Car data-set, the response variable is the Acceptability of Used cars on a scale of unacceptable(1210), acceptable(383), good(69) and very good(65) denoted by 1,2,3 and 4 respectively. The predictors are the buying price(Vhigh, high, Med low), the maintenance price(Vhigh, high, Med low), doors(2,3,4,5,more), persons(2,4,more), boot(small,med,big) and safety(low,med,high). We mention that the response and the covariates are all ordinal. We observe that the class distribution is very skewed and most of the cars are Unacceptable. We do the test-train split and report results for the validation set.

We see from 1 that the results for some partitions are better than others. For example, the results for Ordered partitions (2) and OnevsPrevious (1) are much worse than OnevsNext (3) and OnevsFollowers (5) . In our opinion, this is due to a skewed class distribution. Ordered partitions does not have enough examples of differences for the non-terminal classes and is not able to differentiate them effectively from the terminal ones. Similarly, we see from the partitions of the data used by OnevsFollowers that the classifiers which learn the classification for the under-represented classes.

On the other hand, we see that this issue of unequal classes is less severe for the classifiers that are learnt in OnevsNext and OnevsFollowers. The classes that are underrepresented get combined which overcomes the skewed distribution of classes and improves the predictions.

We also show the confusion matrices for the classifiers without using isotonic regression. We observe that the predictions for OnevsFollowers (5 and 4) were indeed improved after its output was calibrated using isotonic regression. The poor performance in the under-represented classes was nearly improved to perfection. We would also point out that without calibration, the classifiers were not making mistakes with extreme misclassification costs.

On the other hand, we see that the predictions for OnevsPrevious (1 and 7) were badly mangled after calibration. We see that before calibration, the predictions were reasonable, if not perfect and after calibration, the predictor returned one regardless of the input.Essentially, we see that the predictions for the OnevsPrevious model were essentially thrown out of the window by isotonic regression. Finally, we note that that Ordinal regression can work even when the data is heavily imbalanced if the partitioning of the data is done appropriately, such as OnevsFollowers in our case.

Table 1: Confusion Matrices for the Car data-set. Rows correspond to the Truth and the columns correspond to Predictions

Table 2: Ordered Partitions

	1	2	3	4
1	356	0	0	0
2	123	0	0	0
3	18	0	0	0
4	1	0	0	19

Table 3: OnevsNext

	1	2	3	4
1	301	48	7	0
2	79	36	0	8
3	12	0	0	6
4	0	3	0	17

Table 4: OnevsFollowers w/o calibration

	1	2	3	4
1	354	2	0	0
2	3	120	0	0
3	0	18	0	0
4	0	20	0	0

	1	2	3	4
1	355	1	0	0
2	4	118	1	0
3	0	1	17	0
4	0	5	0	15

Table 5: OnevsFollowers

	1	2	3	4
1	356	0	0	0
2	123	0	0	0
3	18	0	0	0
4	20	0	0	0

Table 6: OnevsPrevious

	1	2	3	4
1	180	176	0	0
2	0	119	0	4
3	0	0	0	18
4	0	0	0	20

Table 7: OnevsPrevious w/o calibration

4. Conclusion

In this report we approached ordinal regression firstly using a generalization of the commonly used POM model to fit more cases where the data is not ”nicely” distributed so as to classify it using parallel slopes. For that reason we generated a model which fits a broader spectrum of data distributions: the semi-parallel model. This along with the Elastic-Net Penalty, which by construction borrows Lasso's and Ridge regression qualities, gives us a much better predictions. For a more thorough explanation we prompt the reader to (6). Secondly, we approached it from the classification perspective and incorporated the order information by partitioning the input data . We calibrated the probabilities in various ways and observe that the partitioning can be done cleverly to deal with imbalance in the data. Lastly, we observed that calibration using isotonic regression improved the predictions for the underrepresented classes.

References

[1] Pedro Antonio Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering, 28(1):127–146, 2015.

[2] Peter McCullagh. Regression models for ordinal data. Journal of the Royal Statistical Society: Series B (Methodological), 42(2):109–127, 1980.

[3] Bercedis Peterson and Frank E Harrell Jr. Partial proportional odds models for ordinal response variables. Journal of the Royal Statistical Society: Series C (Applied Statistics), 39(2):205–217, 1990.

[4] UCLA. Ordinal logistic regression — r data analysis examples.

[5] Vladimir N Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5):988–999, 1999.

[6] Michael J Wurm, Paul J Rathouz, and Bret M Hanlon. Regularized ordinal regression and the ordinalnet r package. arXiv preprint arXiv:1706.05003, 2017.

5. Appendix

5.1. Penalties

The penalized objective function is

$$\frac{-1 \times loglik}{n} + penalty$$

The lasso penalty is:

$$\lambda(\rho\|\beta\|_1 + \sum_{k=1}^{K-1} \|\beta_k\|_1)$$

where $\lambda \geq 0$ determines the overall penalty strength and $\rho \geq 0$ determines penalty strength on ordinal coefficients.
The ridge penalty is:

$$\frac{\lambda}{2}(\rho\|\beta\|_2^2 + \sum_{k=1}^{K-1} \|\beta_k\|_2^2)$$

Thus, the elastic net penalty is:

$$\alpha \times Lasso + (1-\alpha) \times Ridge$$

where $\alpha \in [0, 1]$ determines the weighting between lasso and ridge penalty.

5.2. Details about the POM model

The main characteristic of the POM model is that the relationship between each pair of outcome groups is the same. In other words, ordinal logistic regression assumes that the coefficients that describe the relationship between, say, the lowest versus all higher categories of the response variable are the same as those that describe the relationship between the next lowest category and all higher categories, etc. This is called the proportional odds assumption or the parallel regression assumption. Although our data can be ordinal the parallel assumption may not always hold and we may have to use other approaches. But how can we check that assumption holds? There are many ways to examine that hypothesis. Here we state one of them which also provides a visual representation of why the parallel assumption may hold or not.
We can evaluate the parallel slopes assumption by running a series of binary logistic regressions with

varying cutpoints on the dependent variable and checking the equality of coefficients across cutpoints. We thus relax the parallel slopes assumption to checks its tenability.
For information on the exact algorithm we refer the reader to (4).

5.3. Graphs

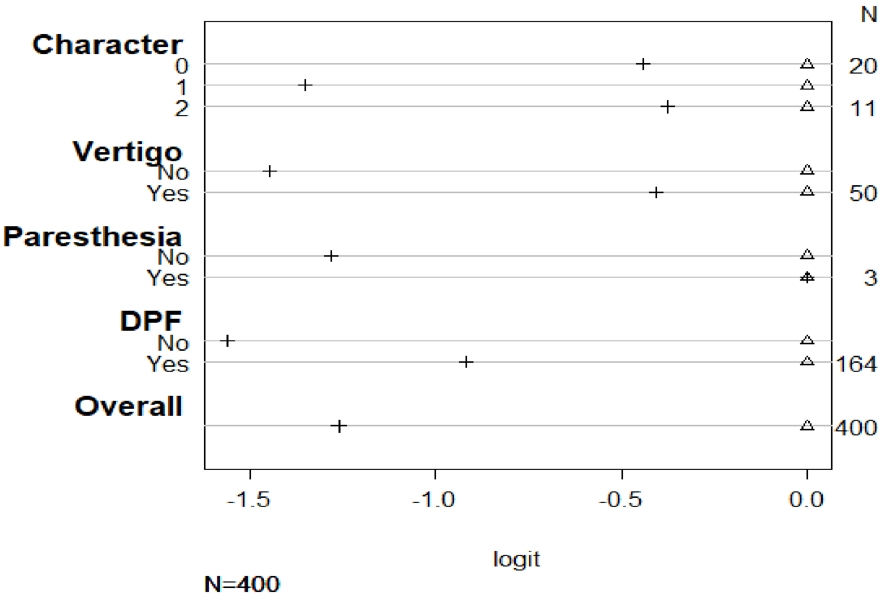

Figure 2: Case where the parallel assumption does not hold: The response variable is the duration of migraines (1,2,3) and some covariates are Character (1,2), Vertigo (0,1), Paresthesia (0,1), DPF(0,1) among 19 others.

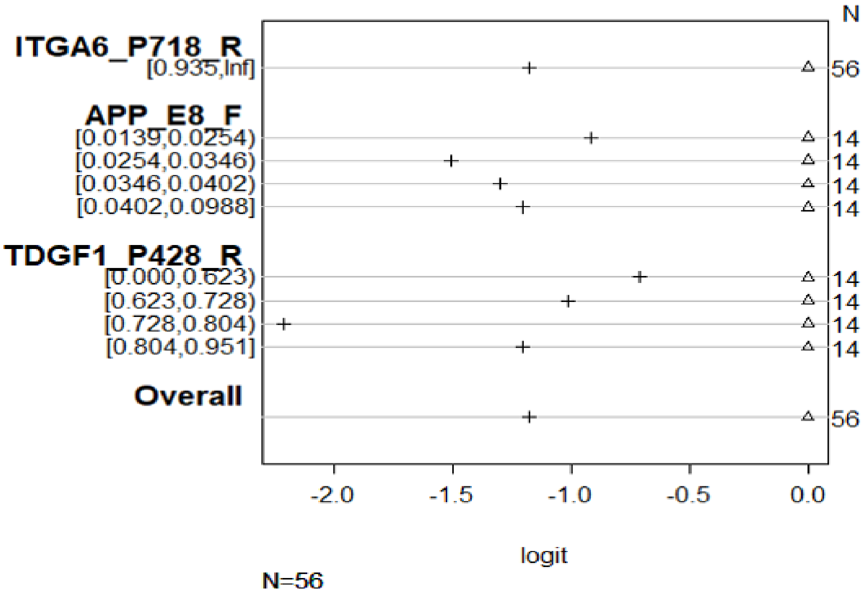

Figure 3: Case where the parallel assumption may hold (data set Hccframe)

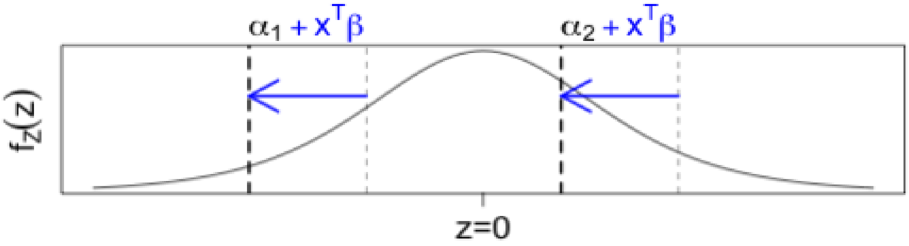

Figure 4: Shifting of $x^T b$ in POM

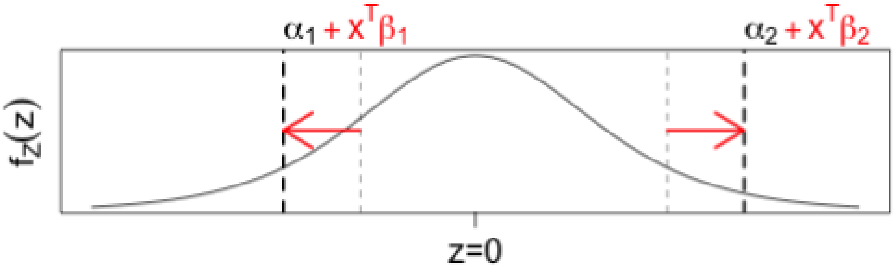

Figure 5: Shifting of $x^T B$ in partial proportional odds model

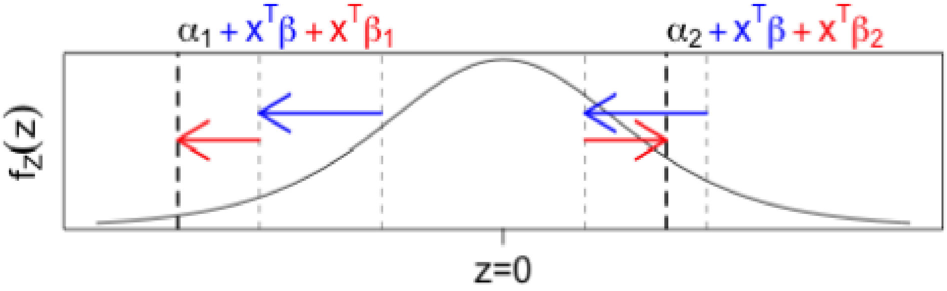
$\alpha_1 + x^T\beta + x^T\beta_1$
$\alpha_2 + x^T\beta + x^T\beta_2$
$f_Z(z)$
z=0

Figure 6: Shifting of $x^T B$ in semi-parallel (blended model)