TRABAJO FIN DE MÁSTER

Máster Inteligencia Computacional e Internet de las Cosas

Implementación de optimización con costes en métodos de regresión ordinal para problemas desbalanceados

Autor: Pedro Juan Torres González
Directores: Javier Sánchez Monedero
David Guijo Rubio

abril, 2024

Índice general

Resumen

La regresión ordinal, también llamada, clasificación ordinal, es un tipo de problema cuyo objetivo es predecir el valor de una variable discreta que sigue una relación de orden. Las particularidades de estos problemas requieren métodos de aprendizaje y métricas de rendimiento específicos. Además, presentan habitualmente problemas como el desbalanceo de clases. Esto es, una o varias clases minoritarias están subrepresentadas en comparación con las clases mayoritarias, lo que en general tiende a producir modelos sesgados hacia las clases mayoritarias y con peor precisión para las clases minoritarias, resultando en modelos poco adecuados en la práctica.

Por otro lado, en clasificación, la regresión logística ha sido y es una herramienta fundamental en la estadística y el aprendizaje automático y, frecuentemente, se sitúa como el mejor modelo en una gran variedad de problemas. Además presenta propiedades importantes para muchos dominios como la eficiencia computacional, la interpretabilidad y el modelado probabilístico.

Existen distintas propuestas de adaptación de la regresión logística a la regresión ordinal, muchas de ellas implementadas en el paquete `mord`, en Python, que está construido siguiendo la interfaz de la conocida biblioteca `scikit-learn`. Sin embargo, al contrario que otras implementaciones de regresión logística, `mord` no implementa mecanismos para afrontar implícitamente el problema del desbalanceo de clases. En particular, los métodos lineales como son el LogisticAT y el LogisticIT, carecen de la funcionalidad para incorporar pesos o costos asociados a cada clase, lo que podría limitar su eficacia y aplicabilidad en situaciones de desbalanceo.

El objetivo de este Trabajo Final de Máster es implementar y evaluar la inclusión de la optimización de los modelos por medio de pesos por clases en la clasificación ordinal con `mord`. Se investigará cómo esto incide en el desempeño de los métodos en entornos desbalanceados, a través de experimentos controlados con diversas bases de datos. El estudio busca proporcionar conocimientos que mejoren la capacidad predictiva y la robustez de los modelos de clasificación ordinal en contextos desbalanceados a través de esta incorporación de costes.

Palabras clave: Optimización, Problemas desbalanceados, Regresión logística, Clasificación ordinal, `mord`, Optimización por pesos de clase.

Abstract

Ordinal regression, also called ordinal classification, is a type of problem where the objective is to predict the value of a discrete ordered variable. The peculiarities of these problems require specific learning methods and performance metrics, and they often present problems such as class imbalance. That is, one or more minority classes are under-represented compared to the majority classes, which generally tends to produce models that are biased towards the majority classes and performs worse for the minority classes, resulting in models that are poorly suited to practice.

On the other hand, logistic regression has been and remains a fundamental tool in statistics and learning for classification, often ranking as the best model in a wide range of problems. It also has important properties for many domains, such as computational efficiency, interpretability, and probabilistic modelling.

Several proposals have been made to adapt logistic regression to ordinal regression. Many of these proposals have been implemented in the Python package `mord`, which follows the API (Application Programming Interface) of the well-known `scikit-learn` software. However, unlike other logistic regression implementations, `mord` does not include mechanisms to explicitly address the problem of class imbalance. In particular, linear approaches such as LogisticAT and LogisticIT lack the functionality to incorporate weights or costs associated with each class, limiting their effectiveness in situations of significant imbalance.

The aim of this Master's Thesis is to implement and evaluate the inclusion of weights in ordinal classification using `mord`. The study will investigate the impact of unbalanced environments on method performance through controlled experiments using various datasets. Its objective is to provide insights that enhance the predictive power and robustness of ordinal classification models in unbalanced contexts thought the inclusion of costs.

Keywords: Optimisation, Imbalanced problems, Logistic regression, Ordinal Classification, `mord`, Class weights optimisation.

Capítulo 1

Introducción

Los modelos de regresión logística han sido ampliamente utilizados en la investigación estadística y en la industria debido a su capacidad para modelar relaciones entre variables de entrada y una variable de respuesta. En trabajos anteriores [1], se exploraron estos modelos de respuesta discreta construyendo el modelo de regresión logística múltiple desde cero y destacando su utilidad en la predicción de resultados no continuos a partir de un conjunto diverso de variables explicativas.

En esta investigación, nos proponemos explorar la aplicación de la regresión logística ordinal en escenarios caracterizados por el desbalanceo de clases. La regresión logística ordinal es una extensión de la regresión logística tradicional que permite clasificar variables categóricas ordenadas en lugar de simplemente distinguir entre dos categorías. Este enfoque ofrece una perspectiva más rica y matizada de los problemas de clasificación, lo que podría traducirse en una mayor precisión y capacidad de generalización en entornos desbalanceados, donde la distribución de las clases objetivo está altamente sesgada, lo que puede afectar negativamente el rendimiento de los modelos de regresión logística. En este contexto, surge la necesidad de abordar este desafío y mejorar la capacidad de los modelos para manejar situaciones de desbalanceo.

Para abordar este desafío, nos basaremos en el trabajo previo del grupo de investigación "Aprendizaje y Redes Neuronales Artificiales" (AYRNA) de la Universidad de Córdoba. A lo largo de los años, el grupo ha demostrado experiencia en abordar problemas desbalanceados en una variedad de contextos [2], incluidos problemas tabulares, de series temporales o de imágenes. Su enfoque en técnicas de computación como redes neuronales artificiales, algoritmos evolutivos y otras metaheurísticas, nos proporciona una base sólida para esta investigación.

Además, nos apoyaremos en el paquete `mord` [3], que implementa varios métodos de clasificación ordinal, incluidos LogisticAT y LogisticIT. Aunque `mord` ofrece herramientas potentes para la clasificación ordinal, carece de la capacidad de introducir pesos o costos asociados a cada clase, una funcionalidad crucial para mitigar el impacto del desbalanceo de clases en la clasificación [4].

Capítulo 2

Objetivos

El objetivo principal de este trabajo es implementar y estudiar el impacto de la inclusión de pesos asociados a cada clase en clasificación ordinal, utilizando el paquete `mord`. Específicamente, se busca:

1. Implementar la inclusión de pesos por clase en el paquete `mord` para los métodos incluidos en este.

2. Evaluar el rendimiento de los métodos de clasificación ordinal implementados en `mord` en comparación con la mejora de la inclusión de pesos. Investigando cómo la inclusión de pesos o costos asociados a cada clase afecta el rendimiento del modelo en problemas de desbalanceo de clases.

3. Realizar experimentos para valorar la contribución de la inclusión de pesos en bases de datos ordinales desbalanceadas. Una forma de obtener problemas ordinales desbalanceados es a partir de problemas de regresión que son discretizados para poder ser enfrentados desde el punto de la clasificación ordinal.

Capítulo 3

Antecedentes

Este trabajo puede considerarse una continuación del Trabajo de Fin de Grado [1] del autor de este Trabajo Fin de Máster, en el cual se construye, analiza y demuestra el modelo de Regresión Logística Múltiple [5]. En este caso, el paradigma a considerar son métodos de clasificación ordinal para la cual contamos con experiencia previa del grupo de investigación AYRNA de la Universidad de Córdoba [6]

3.1. Clasificación desbalanceada

La clasificación desbalanceada es un problema común en problemas de aprendizaje automático sobre el que la comunidad lleva décadas trabajando [7, 8, 9]. A groso modo, podríamos hablar de las siguientes estrategias de modelado de problemas desbalanceados [8]:

1. Pre-procesamiento de datos: técnicas de re-muestreo, aprendizaje activo y uso de pesos en los datos de entrada.
2. Métodos de aprendizaje específicos que ya consideran dicho desbalanceo.
3. Post-procesado de la predicción: métodos de umbral y métodos post-procesado con coste, lo que incluye asociar un coste a la predicción y minimizar el error esperado a este coste.
4. Métodos híbridos.

La técnicas que se implementarán en este TFM se centran en el tercer conjunto de estrategias. El uso de pesos asociados a las clases en problemas de clasificación ordinal no ha sido abordado hasta la fecha, siendo éste un nicho de interés para este tipo de tareas. El riesgo principal a considerar es el sobreajuste, muy común en estos problemas, así como la exploración de una configuración de costes adecuada [8] y se tendrán en cuenta en especial al analizar resultados experimentales.

3.2. Clasificación ordinal

La clasificación ordinal, o regresión ordinal, ya ha sido introducida brevemente como un problema de clasificación supervisada en el que el objetivo es predecir la categoría a la que pertenece un patrón habiendo una relación de orden entre las categorías. La clasificación ordinal se diferencia de la clasificación nominal en que existe una relación de orden entre las categorías; y se diferencia de la regresión estándar en que el número de niveles es finito, y la diferencia entre estos niveles no está definida. De esta forma, la clasificación ordinal se sitúa entre la clasificación nominal y la regresión.

Los métodos específicos de clasificación se centran en aprovechar esta información a priori sobre el orden de las etiquetas, también esperado en el espacio de entrada, y minimizar errores de métricas ordinales específicas. Estas métricas de rendimiento ordinal empleadas consideran el orden de las clases, de tal forma que los errores de clasificación entre clases adyacentes deben ser considerados con menos importancia, que errores de clasificación entre clases no adyacentes (más separadas en la escala ordinal). Por ejemplo, consideremos un conjunto de datos de predicción del tiempo con la variable objetivo tomando valores en el conjunto $\{$*muy frío*, *frío*, *templado*, *caluroso*, *muy caluroso*$\}$, con una clara relación de orden natural entre las clases, *muy frío* $\prec$ *frío* $\prec$ *templado* $\prec$ *caluroso* $\prec$ *muy caluroso*. Es evidente que predecir erróneamente la clase *caluroso* cuando la clase real es *frío* representa un error más grave que el error asociado a predecir *muy frío* [10, 11].

La Figura 3.1 muestra diferentes tipos de métodos de regresión ordinal. Este TFM se centrará en métodos de regresión logística ordinales, que se incluyen en la categoría de métodos de umbral (*threshold models*).

3.3. Software libre para clasificación ordinal y problemas desbalanceados

El grupo de investigación AYRNA mantiene el proyecto de software libre `ORCA` [12] que recoge numerosos métodos y métricas específicos para clasificación ordinal. `ORCA` está desarrollado en MATLAB/Octave lo que dificulta, a día de hoy, su uso y adopción y es por ello que el grupo AYRNA trabaja en portar muchas de sus funcionalidades al entorno Python basado en `scikit-learn` [13].

En este contexto, surge el paquete `mord` [14] que agrupa varios métodos de clasificación ordinal en Python basados en regresión logística y regresión *ridge* [1]. Sin embargo, `mord` carece de la funcionalidad para incorporar pesos o costos asociados a

[1] `http://github.com/fabianp/mord`

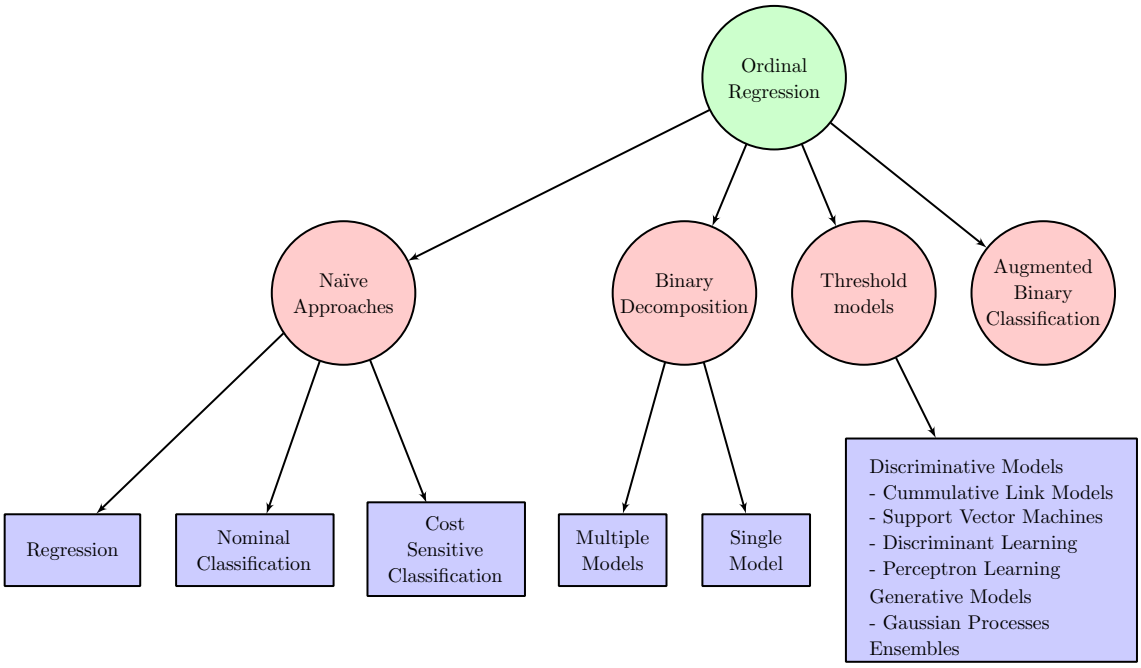

Figura 3.1: Taxonomía de métodos de regresión ordinal. Fuente [6].

cada clase, lo que podría afectar su eficacia en situaciones de desbalanceo. Además de estos entornos, existe el proyecto `imbalanced-learn` [15], que implementa muchas métricas de evaluación de situaciones desbalanceadas así como técnicas de remuestreo de bases de datos.

Este TFM se inspirará en los proyectos anteriores para implementar el aprendizaje ordinal con costes y evaluar su rendimiento en dominios desbalanceados. Los métodos serán construidos a partir de la implementación actual de `mord`.

Capítulo 4

Partes del Proyecto

El proyecto se organiza en las siguientes etapas detalladas:

- **Implementación y evaluación de métodos de clasificación ordinal:** se realizará la implementación y evaluación de los métodos LogisticAT y LogisticIT del paquete `mord`. Estos métodos se aplicarán en conjuntos de datos de clasificación ordinal para analizar su rendimiento y capacidad de generalización.
- **Evaluación del impacto de pesos en la clasificación ordinal:** se llevará a cabo un estudio exhaustivo para examinar cómo la inclusión de pesos o costos asociados a cada clase afecta el rendimiento de los modelos implementados, especialmente en situaciones de desbalanceo de clases. Se evaluará la capacidad de los métodos para abordar eficazmente este desafío.
- **Experimentación con bases de datos desbalanceadas:** se emplearán varias bases de datos [2], particularmente aquellas con un desbalanceo de clases pronunciado, para evaluar la efectividad de los métodos implementados en comparación con el Proportional Odd Model (POM) sin costos asociados, equivalente al método estándar de regresión logística para clasificación ordinal. Se llevarán a cabo experimentos para medir la precisión y la robustez de los modelos en diferentes escenarios.
- **Análisis de la discretización de datos continuos:** Se realizará un análisis exhaustivo del impacto de la discretización de datos continuos utilizando KBinsDiscretizer de `scikit-learn` en clasificación ordinal. Se explorará su capacidad para manejar desbalanceo de clases y se comparará con otros enfoques de discretización.

Estas etapas del proyecto se llevarán a cabo mediante experimentos controlados

y análisis estadísticos detallados para evaluar el rendimiento y la eficacia de los métodos propuestos.

4.1. Planificación Temporal

Las fases de desarrollo del proyecto serán las siguientes:

- Estudio y análisis del problema (60 horas).
- Diseño (60 horas).
- Implementación (90 horas).
- Pruebas y experimentación (80 horas).
- Documentación del proyecto (60 horas).

Todas estas fases del proyecto suman:

- 350 horas.

Capítulo 5

Recursos

Para la realización del proyecto se dispondrán de los siguientes recursos:

5.1. Recursos humanos

Autor: Pedro Juan Torres González
Alumno del Máster en Inteligencia Computacional e Internet de las Cosas.

- **Email**: z32togop@uco.es
- **DNI**: 31029947-A
- **Titulación**: Máster en Inteligencia Computacional e Internet de las Cosas

Director: Dr. Javier Sanchez Monedero.
Doctor en Ciencias de la Computación e Inteligencia Artificial por la Universidad de Granada. Será encargado de dar soporte técnico y teórico en el estudio, diseño e implementación del TFM.

- Investigador senior Beatríz Galindo
- **Departamento**: Informática y Análisis Numérico
- Escuela Politécnica Superior de Córdoba
- Universidad de Córdoba
- **Email**: jsanchezm@uco.es

Director: Dr. David Guijo Rubio.
Doctor en Ciencias de la Computación e Inteligencia Artificial por la Universidad de Córdoba. Será encargado de dar soporte técnico y teórico en el estudio, diseño e implementación del TFM.

- Investigador Juan de la Cierva
- **Departamento**: Teoría de la señal y las comunicaciones
- Escuela Politécnica Superior
- Universidad de Alcalá
- **Email**: david.guijo@uah.es

5.2. Recursos materiales

5.2.1. Recursos software

- **Ubuntu 19.10 o superiores** [16]: sistema operativo para la programación, prueba de la herramienta y generación de la documentación.
- **Python** [17]: lenguaje de programación.
- **Git** [18]: programa para la gestión de versiones.
- **Github** [19]: plataforma para acceder al código y a sus distintas versiones en la nube.
- **LaTeX** [20]: sistema de composición tipográfica para la realización de la documentación.
- **Overleaf** [21]: editor en línea de LaTeX.

5.2.2. Recursos hardware

Para la realización de la programación del proyecto se hará uso del ordenador personal del proyectista, que se trata de un *LENOVO IdeaPad Slim 3 15IAH8* el cuál tiene las siguientes características:

- Procesador 12th Gen Intel(R) Core(TM) i5-12450H
- Memoria RAM de 16 GB.

- Memoria SSD 512 GB.
- Gráfica Intel(R) UHD Graphics for 12th Gen Intel® Processors.

En cuanto a la realización de las pruebas y la experimentación, se contará con el uso del clúster de computación del grupo de investigación AYRNA, que cuenta con aproximadamente 600 núcleos de procesamiento y 2 TB de memoria. Todos los procesos experimentales se realizan a través del entorno de computación de alto rendimiento HTCondor [22]. Además, el clúster cuenta con 28 GPUs NVIDIA dedicadas específicamente a la realización de experimentos de Deep Learning.

Bibliografía

[1] Pedro J Torres González. Aspectos matemáticos de la regresión logística múltiple. 2023.

[2] Pedro Antonio Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervás-Martínez. [DATASETS]-Ordinal regression methods: Survey and Experimental Study. URL `https://www.uco.es/grupos/ayrna/orreview`.

[3] Fabian Pedregosa-Izquierdo. *Feature extraction and supervised learning on fMRI : from practice to theory*. phdthesis, Université Pierre et Marie Curie - Paris VI, February 2015. URL `https://theses.hal.science/tel-01100921`.

[4] Thomas Oommen, Laurie G. Baise, and Richard M. Vogel. Sampling Bias and Class Imbalance in Maximum-likelihood Logistic Regression. *Mathematical Geosciences*, 43(1):99–120, January 2011. ISSN 1874-8953. doi: 10.1007/s11004-010-9311-8. URL `https://doi.org/10.1007/s11004-010-9311-8`.

[5] David W. Hosmer, Jr, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. John Wiley & Sons, 4 2013.

[6] Pedro Antonio Gutierrez, 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, January 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2015.2457911. URL `http://ieeexplore.ieee.org/document/7161338/`.

[7] Haibo He and Edwardo A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009. ISSN 1558-2191. doi: 10.1109/TKDE.2008.239. URL `https://ieeexplore.ieee.org/document/5128907`. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[8] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2), aug 2016. ISSN 0360-0300. doi: 10.1145/2907070. URL https://doi.org/10.1145/2907070. Available at http://arxiv.org/abs/1505.01658.

[9] Yang Feng, Min Zhou, and Xin Tong. Imbalanced classification: a paradigm-based review, June 2021. URL http://arxiv.org/abs/2002.04592. arXiv:2002.04592 [stat].

[10] Joaquim F. Pinto da Costa, Hugo Alonso, and Jaime S. Cardoso. The unimodal model for the classification of ordinal data. *Neural Networks*, 21:78–91, January 2008. ISSN 0893-6080.

[11] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P.A. Gutiérrez. Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing*, 135:21–31, July 2014. ISSN 09252312. doi: 10.1016/j.neucom.2013.05.058. URL https://linkinghub.elsevier.com/retrieve/pii/S0925231213011399.

[12] Javier Sánchez-Monedero, Pedro A. Gutiérrez, and María Pérez-Ortiz. Orca: A matlab/octave toolbox for ordinal regression. *Journal of Machine Learning Research*, 20(125):1–5, 2019. URL http://jmlr.org/papers/v20/18-349.html.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

[14] Fabian Pedregosa-Izquierdo. *Feature extraction and supervised learning on fMRI : from practice to theory.* Theses, Université Pierre et Marie Curie - Paris VI, February 2015. URL https://theses.hal.science/tel-01100921.

[15] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

[16] Ubuntu. https://ubuntu.com/. Último acceso: 02-04-2024.

[17] Python. https://www.python.org/. Último acceso: 02-04-2024.

[18] Git –distributed-is-the-new-centralized. https://git-scm.com/, .

[19] GitHub. https://github.com/, . Último acceso: 02-04-2024.

[20] LaTeX – A document preparation system. https://www.latex-project.org/. Último acceso: 02-04-2024.

[21] Overleaf. https://www.overleaf.com/. Último acceso: 02-04-2024.

[22] HTCondor Software Suite. http://research.cs.wisc.edu/htcondor/.