ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA
Universidad de Córdoba

PROPUESTA DE TEMA DE TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Aprendizaje multi-instancia multi-etiqueta con scikit-learn

Autor
Damián Martínez Ávila
DNI: 46271668S

Directora
Dª. Eva Lucrecia Gibaja Galindo

Octubre, 2023

UNIVERSIDAD D CORDOBA

CONTENIDO

1. INTRODUCCIÓN

La clasificación tradicional en el ámbito del aprendizaje automático se trata de un enfoque del aprendizaje supervisado, el cual consiste en estudiar una correspondencia entre un conjunto de variables de entrada X y una variable de salida Y y analizar esta correspondencia para predecir las salidas de datos no observados. [1]

Cuando hablamos de la estructura de datos usada, la información está dividida en patrones. En la clasificación convencional, cada patrón está representado por una única instancia con un número determinado de atributos y una sola etiqueta.

x1	x2	x3	x4	x5	x6	y

Aunque la formalización anterior prevalece y tiene éxito, hay muchos problemas del mundo real que no se ajustan bien a esta representación, en los que un objeto del mundo real puede estar asociado a varias etiquetas simultáneamente [2].

Este grupo de problemas representa un área conocida como clasificación multi-etiqueta [3]. La mayoría de los trabajos sobre clasificación multi-etiqueta comenzaron como un intento de tratar las ambigüedades encontradas en los problemas de clasificación de documentos.

x1	x2	x3	x4	x5	x6	y1	y2	y3

El aprendizaje multi-instancia es una variante del aprendizaje supervisado que consiste en aprender un concepto a partir de bolsas de instancias positivas y negativas. Cada bolsa puede contener muchas instancias, pero una bolsa se etiqueta como positiva aunque sólo una de las instancias que contiene corresponda al concepto. Una bolsa se etiqueta como negativa sólo si todas las instancias que contiene son negativas. [4]

x1	x2	x3	x4	x5	x6	
x1	x2	x3	x4	x5	x6	y
x1	x2	x3	x4	x5	x6	

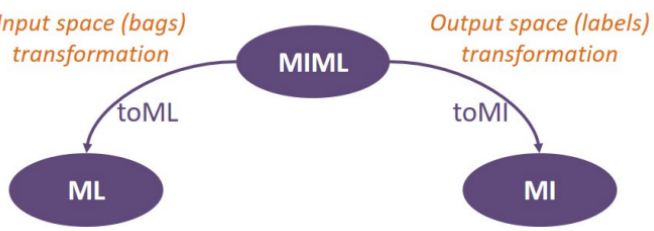
De este modo, el aprendizaje con múltiples instancias y múltiples etiquetas combina ambos marcos de aprendizaje para introducir una mayor flexibilidad tanto en la representación de la entrada, como en las salidas. Cada objeto es representado por una bolsa de instancias y se le permite tener múltiples etiquetas de clase.

x1	x2	x3	x4	x5	x6	y1	y2	y3
x1	x2	x3	x4	x5	x6			
x1	x2	x3	x4	x5	x6			
x1	x2	x3	x4	x5	x6	y1	y2	y3
x1	x2	x3	x4	x5	x6			
x1	x2	x3	x4	x5	x6	y1	y2	y3
x1	x2	x3	x4	x5	x6			
x1	x2	x3	x4	x5	x6			

Este aprendizaje permite formalizar objetos en diferentes problemas complejos. Por ejemplo, una imagen contiene de forma natural diferentes regiones (instancias) y la imagen completa puede representar diferentes clases, tales como nubes, leones, y paisajes. En categorización de textos, cada documento normalmente consiste en diferentes secciones o párrafos (instancias), y cada documento puede ser asignado a diferentes ítems, tales como deportes, política, y ocio.

Figura 1: Clasificación Multi-etiqueta Multi-instancia

Se plantea profundizar en este marco de aprendizaje desarrollando una librería que permita trabajar con modelos que resuelvan este problema. Como el aprendizaje MIML se basa tanto en el aprendizaje MI como en el aprendizaje ML, se pueden aplicar dos tipos de transformaciones para resolver un problema MIML. El problema se puede transformar en un problema MI y resolverse mediante algoritmos MI o transformarlo en un problema ML y resolverlo mediante algoritmos ML [5].



2. ANTECEDENTES

En este TFG se va a trabajar con las siguientes herramientas:

Numpy: NumPy es una biblioteca de código abierto de Python que se utiliza en casi todos los campos de la ciencia y la ingeniería. Es el estándar universal para trabajar con datos numéricos en Python, y está en el núcleo de los ecosistemas científicos Python y PyData. La API de NumPy se utiliza ampliamente en Pandas, SciPy, Matplotlib, scikit-learn, scikit-image y la mayoría de los paquetes científicos y de ciencia de datos de Python [6].

SciPy: SciPy es una colección de algoritmos matemáticos y funciones prácticas basadas en NumPy . Añade una potencia significativa a Python al proporcionar al usuario comandos y clases de alto nivel para manipular y visualizar datos [7].

Scikit Learn: scikit-learn es una biblioteca de aprendizaje automático de software libre para el lenguaje de programación Python. Incluye varios algoritmos de clasificación, regresión y agrupación, como máquinas de vectores soporte, bosques aleatorios, gradient boosting, k-means y DBSCAN, y está diseñada para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy de Python.

Cabe citar la librería MIML escrita en Java cuyo objetivo es facilitar el desarrollo, testeo y comparación de algoritmos de clasificación para el aprendizaje multi-instancia multi-etiqueta (MIML).

Incluye 43 algoritmos de clasificación para resolver problemas MIML. Se incluyen algoritmos sobre tres enfoques diferentes para resolver un problema MIML: transformar el problema a multi-instancia, transformar el problema a multi-etiqueta y resolver directamente el problema MIML. Además, proporciona procedimientos de validación cruzada y de espera, métricas estándar para la evaluación del rendimiento, así como la generación de informes [8].

3. OBJETIVOS

El objetivo principal de este Trabajo Fin de Grado es desarrollar una librería en Python que nos permita trabajar y resolver problemas utilizando el aprendizaje con múltiples instancias y múltiples etiquetas.

Más concretamente, los objetivos que se pretenden alcanzar con la realización de dicho proyecto serían los siguientes:

Estudio de la estructura de clases de las librerías a utilizar. De este modo, la librería que se pretende desarrollar hará uso de las funcionalidades necesarias de dichas librerías. En el caso que fuese necesario, se adaptarían algunas funcionalidades de ellas para resolver nuestro problema.

Poder conseguir un manejo básico de datasets MIML en formato arff que permita realizar tareas entre las que se incluyen:

- Cargar los datasets en memoria para su posterior procesamiento.
- Particionado para validación cruzada de k iteraciones sobre los datos.
- Obtener métricas sobre las características más relevantes de los datasets (número de patrones, número medio de instancias por patrón, número medio de etiquetas por patrón, etc.).

Capacidad para reducir la dificultad de los problemas de la clasificación MIML mediante el uso de transformaciones básicas de los datos (como la aritmética, geométrica y min-max), para que se resuelvan como un caso de clasificación multi-etiqueta.

4. FASES DE DESARROLLO DEL PROYECTO

El Trabajo Fin de Grado se desarrollará en las siguientes etapas:

Estudio e investigación: Se estudiarán las distintas tecnologías necesarias para el desarrollo del Trabajo Fin de Grado y se investigará en profundidad acerca del tema de aprendizaje MIML.

Análisis y definición de requisitos: Se analizará lo anteriormente investigado para obtener los requisitos del proyecto.

Diseño: Se definirá la estructura para el software asegurandose de que cumplirá los requisitos previamente establecidos.

Implementación: Se traducirá el diseño y la planificación previos en código ejecutable y funcional, en el lenguaje de programación Python.

Pruebas: Comprobación de la calidad, fiabilidad y funcionamiento correcto de la librería.

Documentación: Se documentará cada una de las etapas a lo largo de todo el desarrollo del trabajo.

5. RECURSOS

5.1. Recursos Humanos

El Trabajo Fin de Grado será realizado por Damián Martínez Ávila con la supervisión de la profesora Eva Lucrecia Gibaja Galindo.

5.2. Recursos Software

Para llevar a cabo la implementación del módulo desarrollado se utilizará el lenguaje Python. De este modo se aprovecharán las funcionalidades de librerías como Scikit Learn, Numpy y Pandas proporcionan para el procesamiento de los datos.

El entorno de programación que se utilizará será Visual Studio Code.

La documentación se realizará con Latex.

5.3. Recursos Hardware

Como equipo hardware para el desarrollo se ha utilizado un ordenador portátil personal con las siguientes características:

- Procesador Ryzen 7 a 1.8 GHz
- Memoria RAM 16 GB
- Disco Duro SSD de 512 GB
- Sistema operativo: Windows 11 de 64 bits

6. DISTRIBUCIÓN TEMPORAL DEL PROYECTO

El Trabajo Fin de Grado ha de tener una duración de 300 horas (correspondientes a 12 créditos ECTS correspondientes por completo al alumno).

Para ello hemos realizado una distribución en 15 semanas, con 20 horas de trabajo planificadas para cada semana que se detalla en la Tabla 1.

Semana	Tarea
1-2	Estudio aprendizaje Multi-Etiqueta y aprendizaje Multi-Instancia
3-4	Estudio de Python y las diferentes librerías a utilizar
5-6	Análisis y definición de requisitos
7-9	Diseño y desarrollo de las clases de la librería que se va a implementar
10-11	Implementación de la librería
12-13	Pruebas del software de la librería a realizar
14-15	Reunificación de la documentación realizadas en cada una de las fases

Cuadro 1: Distribución temporal del proyecto

BIBLIOGRAFÍA

[1] P. Cunningham, M. Cord, and S. J. Delany, *Supervised Learning*, pp. 21–22. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[2] A. R. Rout, “A deep learning model for image classification,” *IRJET*, 2017.

[3] A. C. P. L. F. de Carvalho and A. A. Freitas, *A Tutorial on Multi-label Classification Techniques*, pp. 177–195. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[4] O. Maron and T. Lozano-Pérez, “A framework for multiple-instance learning,” in *Advances in Neural Information Processing Systems* (M. Jordan, M. Kearns, and S. Solla, eds.), vol. 10, MIT Press, 1997.

[5] A. Z. G. Álvaro Belmonte Pérez, Eva Gibaja Galindo, “Miml library user guide,” 2023.

[6] “Numpy documentation,” 2023.

[7] “Scipy documentation,” 2023.

[8] A. Z. G. Álvaro Belmonte Pérez, Eva Gibaja Galindo, “Miml library,” 2023.