Métodos no supervisados y detección de anomalías: introducción al aprendizaje no supervisado

UCO ONLINE

Introducción al aprendizaje no supervisado

1. Introducción al aprendizaje automático

El aprendizaje automático (ML, *machine learning*), aglutina un conjunto de técnicas computacionales que permiten analizar grandes cantidades de datos para tratar de detectar automáticamente patrones. Estos patrones pueden utilizarse para resolver diferentes tareas, desde etiquetar correos electrónicos como spam, hasta reconocer comandos de voz o expresiones faciales en nuestros dispositivos móviles. En todas estas situaciones, no existe un programa informático que ejecuta unas instrucciones previamente definidas. Por el contrario, el sistema informático adquiere la capacidad de resolver la tarea mediante un proceso similar a cómo los humanos aprendemos, esto es, basándose en ejemplos y la experiencia previa.

Existen dos características esenciales que nos indican que un problema necesita de la aplicación de aprendizaje automático. En primer lugar, debe tratarse de una tarea *compleja de programar*. Este tipo de tareas se corresponde habitualmente con actividades que nosotros, como humanos, hacemos de forma natural como reconocer formas, hablar, movernos, etc. Otro tipo de tareas difíciles de resolver son aquellas que necesitan analizar grandes cantidades de datos, lo cual está fuera de nuestra capacidad cerebral. La segunda característica es la *capacidad de adaptación*. A diferencia de un programa informático, que una vez compilado o instalado, permanece inalterado, los sistemas basados en aprendizaje automático son capaces de adaptarse. Esta adaptación es posible porque el sistema no sigue unas "instrucciones" prefijadas, sino que da una respuesta en base a los modelos que ha aprendido y estos, a su vez, pueden modificarse para ajustarse a nuevos ejemplos observados.

El aprendizaje automático es un concepto amplio que ha ido evolucionando, desde el uso de técnicas estadísticas tradicionales a paradigmas más avanzados, lo cual ha permitido a su vez abordar una mayor diversidad de tareas. Además de la variedad de técnicas, el campo del aprendizaje automático puede clasificarse según varias características que determinan el enfoque con el que se realiza el aprendizaje. A continuación, se presenta una breve categorización de los principales enfoques, lo cual nos permitirá ubicar el paradigma del aprendizaje no supervisado, objeto de la siguiente sección:

- Según la naturaleza de los datos, podemos distinguir entre *aprendizaje supervisado* y *no supervisado*. En el primer caso, el aprendizaje se basa en analizar ejemplos que están *etiquetados*, de forma que el objetivo es predecir esa etiqueta para cualquier nuevo ejemplo que se presentase. Tareas como la regresión y la clasificación se resuelven con este enfoque, tal y como se estudia en la asignatura "Métodos predictivos". En contraposición, el aprendizaje no supervisado no asume la existencia de categorías en los ejemplos, sino que trata de descubrir cuáles pueden ser esas categorías. A menudo, las tareas que se resuelven mediante aprendizaje no supervisado tienen un carácter *exploratorio*. Desde esta perspectiva de los datos, también se suele incluir un tercer paradigma: el *aprendizaje por refuerzo*. En este tipo de enfoque, el aprendizaje se produce en base a acciones que se realizan sobre los datos, en lugar de sobre los datos en sí. Las acciones que ayudan a resolver la tarea son las que "refuerzan" el proceso de aprendizaje.
- En base a cómo se aborda el aprendizaje, se establece una segunda categorización entre aprendizaje *pasivo* y *activo*. El aprendizaje pasivo es la forma tradicional de aprendizaje, donde inicialmente se dispone de toda la información a partir de la cual se aprende (por ejemplo, el conjunto completo de datos etiquetado). En su lugar, el aprendizaje activo fomenta que el sistema interactúe con su entorno para recibir información durante el proceso de aprendizaje. De esta forma, se consigue una mejor adaptación pues se introduce nuevo conocimiento progresivamente. Un caso habitual es disponer de un "oráculo" capaz de proporcionar las etiquetas de los datos cuando no están disponibles, asumiendo que se pueden obtener algunas a un coste determinado. A medio camino entre aprendizaje pasivo y activo encontramos el aprendizaje *semi-supervisado*, donde el proceso de aprendizaje se basa en disponer desde el comienzo de solo una parte de los datos etiquetados.
- Según la forma en la que se controla el proceso de aprendizaje, se puede hablar de aprendizaje estadístico o adversario. En el primero, se sigue la filosofía tradicional de que los datos de entramiento responden a un fenómeno generado por un proceso aleatorio cuya naturaleza se quiere explicar. En el aprendizaje adversario, el sistema aprende mediante el "enfrentamiento" a otra entidad que intenta obstruir el proceso. Al haber aprendido en un entorno "hostil", el sistema estará preparado para ser exitoso en cualquier otro escenario.
- Según el tipo de respuesta que se espere, se puede necesitar un aprendizaje de tipo "por lotes" (*batch*) o en línea (*online*). El primero parte del supuesto de que se dispone de un histórico de datos de los cuales aprender antes de comenzar a tomar decisiones. Es el modelo habitual cuando se dispone de un conjunto de datos grande o que no cambia con frecuencia. Por el contrario, el aprendizaje en línea es necesario cuando los datos se generan y deben procesar en poco tiempo, como cuando son generados por sensores. En esta situación, el aprendizaje se va realizando a medida que se recogen nuevos datos.

2. Características del aprendizaje no supervisado

De la sección anterior podemos destacar que el aprendizaje no supervisado es un enfoque alternativo al aprendizaje supervisado. En aprendizaje supervisado, los datos están perfectamente identificados, esto es, a cada ejemplo se le ha asociado una etiqueta o valor de salida. Esta característica implica que el algoritmo que entrena con esos datos solo va a ser capaz de aprender a distinguir los conceptos asociados a esas etiquetas, pues es lo único que existe en ese conjunto de datos. La tarea que acomete el algoritmo está, por tanto, muy delimitada. En el ejemplo del filtro *anti-spam*, solo seremos capaces de diferenciar los correos electrónicos estándar (sean como sean) de los que son *spam*.

En el aprendizaje no supervisado no existen etiquetas que guíen al algoritmo para saber si está identificando bien un concepto de otro. La tarea que hay que resolver se vuelve por tanto más difusa, pero a la vez está abierta a descubrir cualquier concepto que subyace a los datos, aunque a priori no se sepa que está ahí. Siguiendo con el ejemplo anterior, podríamos darle a nuestro algoritmo una muestra de correos electrónicos, y dejar que sea el algoritmo quien descubra una agrupación de esos correos. Podríamos encontrar correos agrupados según su temática (promociones, notificaciones, noticias, etc.) o agrupados por su origen (trabajo, personal, vinculado a aplicaciones o redes sociales). También podríamos separar correo deseado (mayoritario), de correo no deseado (*spam*) o de correo "sospechoso". Sin establecer estas categorías a priori (lo que serían las etiquetas en aprendizaje no supervisado), existe una mayor libertad a la hora de aprender conceptos sobre los datos. Es más, este tipo de algoritmo sería capaz de identificar nuevos conceptos que apareciesen en el futuro.

Al tener un panorama más abierto, la tarea a resolver no está tan claramente definida como lo estaba en el aprendizaje supervisado. Por este motivo, el aprendizaje no supervisado tiene un carácter claramente exploratorio, donde el objetivo el adquirir conocimiento sobre los datos o identificar las propiedades que diferencian a unos ejemplos de otros. Por este motivo, también se le suele llamar aprendizaje "representacional", ya que lo que se aprende de los ejemplos es un conjunto de propiedades que permiten identificar los distintos tipos de ejemplos. En nuestro ejemplo del correo electrónico, las propiedades serían la temática, el origen o la intencionalidad de los correos electrónicos.

Aunque podemos tener cierta intuición de qué representan nuestros datos, no tenemos la solución "real" con la que comparar si el algoritmo acierta o no. No obstante, en ciertas situaciones puede interesar validar el resultado obtenido, por ejemplo, por medio de un experto que compruebe la correcta identificación de los conceptos. Otra circunstancia que suele verse favorecida por el uso de aprendizaje no supervisado es la escasez de datos. El aprendizaje supervisado es una herramienta potente solo si se dispone de un conjunto de datos grande y bien etiquetado. El aprendizaje no supervisado es una alternativa cuando estas circunstancias no se garantizan.

3. Tareas propias del aprendizaje no supervisado

En esta sección se presentan las principales tareas que se suelen abordar desde la perspectiva del aprendizaje no supervisado. Las más relevantes se estudiarán en profundidad a lo largo de este curso, definiéndolas formalmente y estudiando una variedad de algoritmos para resolverlas.

3.1. Reducción de dimensionalidad

La tarea de reducción de la dimensionalidad consiste en disminuir el número de características que componen un conjunto de datos, manteniendo solo aquellas más importantes. Esta tarea suele abordarse en una fase de preprocesado con el objetivo de reducir la complejidad del conjunto de datos original. Tras aplicar la reducción, pueden ejecutarse otros métodos de aprendizaje tanto supervisado como no supervisado para extraer conocimiento. Es una forma habitual de reducir el coste computacional de la fase de entrenamiento, sobre todo cuando los conjuntos de datos son de alta dimensionalidad (número elevado de características) o contienen datos que son difíciles de procesar, como imágenes o texto. Además, al estar basados en un número menor de características, se consigue obtener modelos de decisión más interpretables por lo general.

Los métodos de reducción de la dimensionalidad se basan en realizar proyecciones lineales o no lineales del conjunto de características original. Dentro de los métodos de proyección lineal destacan PCA (*Principal Component Analysis*), SVD (*Singular Value Decomposition*) y LDA (*Latent Dirichlet Allocation*).

3.2. Agrupamiento

La tarea de agrupamiento tiene como objetivo dividir el conjunto de ejemplos en grupos más pequeños, de forma que los ejemplos en un mismo grupo compartan propiedades. A su vez, los ejemplos distribuidos en distintos grupos deberían ser lo más distintos posibles. La tarea de agrupamiento se basa en detectar similitudes entre los ejemplos, para lo cual se necesita definir qué significa la "similitud" en el contexto concreto del problema a resolver. Un problema habitual es el agrupamiento es decidir a priori cuántos grupos deben identificarse, aunque existen algoritmos capaces de determinar ese número por sí mismos.

El problema de agrupamiento y los métodos para resolverlo serán estudiados de forma detallada en las tres primeras semanas del curso. Aunque se estudiará en detalle más adelante, se puede nombrar aquí el algoritmo k-medias como el método de agrupamiento más popular.

3.3. Detección de anomalías

La detección de anomalías consiste en identificar eventos que no se corresponden con aquello que se espera. Asumiendo que un el fenómeno bajo estudio sigue un procedimiento estacionario, los datos anómalos son aquellos que se "escapan" del comportamiento normal. Por ejemplo, si estamos recogiendo datos de un sensor de temperatura de forma constante, un cambio brusco en los valores de la temperatura puede ser identificado como un comportamiento anómalo. Determinar el grado en que un dato es anómalo es complejo, pues lo que es "extraño" en un momento determinado, puede no serlo en el contexto global del fenómeno en estudio.

Como se verá en la segunda mitad de este curso, la detección de anomalías puede abordarse con varios tipos de métodos, desde técnicas estadísticas hasta aprendizaje automático tanto supervisado como no supervisado.

3.4. Extracción de características

La extracción de características busca crear un nuevo espacio de características con las que representar a los datos. Comparte con la reducción de dimensionalidad la posibilidad de reducir el número de características, pero también tiene la capacidad de generar otras nuevas. Los *autoenconders* son un tipo de redes neuronales profundas que permiten realizar este tipo de transformación en los datos.

Dentro de esta categoría también se pueden mencionar a los mapas autoorganizados o SOM (*self-organizing maps*). En este caso, el objetivo es generar una representación de las características de los datos en un nuevo espacio de tipo discreto, que además es fácil de visualizar gráficamente. Por ejemplo, los SOM son útiles para resumir un texto o documento en un conjunto limitado de temas sobre los que trata. Los SOM son generados por un tipo especial de redes neuronales, y comparte ideas con las tareas de agrupamiento y la reducción de dimensionalidad.

4. Librerías software para aprendizaje no supervisado

En esta sección se presentan algunas librerías software que implementan métodos de aprendizaje no supervisado para abordar las tareas mencionadas anteriormente. En primer lugar, se introducen las librerías desarrolladas en lenguaje Python, que será el que se utilizará principalmente en este curso. A continuación, se enumeran paquetes disponibles en R con funcionalidad equivalente. Finalmente, se describen algunas herramientas de uso libre o comercial que también incluyen algoritmos de aprendizaje no supervisado.

4.1. Librerías para aprendizaje no supervisado en Python

Python es uno de los lenguajes de programación más utilizados para hacer desarrollos y experimentos en aprendizaje automático. Centrándonos en el aprendizaje no supervisado, cabe destacar los siguientes recursos software:

- ***Scikit-learn*** es la librería en Python por excelencia para ejecutar algoritmos de aprendizaje automático. Dentro de sus paquetes, incluye métodos de aprendizaje no supervisado con los que abordar los problemas de agrupamiento, reducción de la dimensionalidad o detección de valores anómalos. Su desarrollo comienza en 2007, haciéndose público en 2010. Recientemente se ha lanzado la versión 1.0.1 (octubre 2021). Esta librería se fundamenta en otras de gran uso como son *NumPy*, *SciPy* y *matplotlib*. A lo largo del curso, se utilizará esta librería para ejemplos y tareas de asignación, por lo que se aconseja conocer su documentación: https://scikit-learn.org/
- ***Pyclustering*** es una librería Python especializada en métodos de agrupamiento. Internamente, incluye algunas implementaciones en C++ para mejorar su eficiencia. El proyecto es también de código abierto y está disponible desde 2016. En la siguiente dirección web pueden consultarse sus funcionalidades: https://pyclustering.github.io/
- ***PyOD*** es una *toolkit* para el análisis de datos que implementa métodos para la detección de *outliers* y anomalías. El proyecto comenzó a desarrollarse en 2017 y actualmente cuenta con más de 30 algoritmos disponibles, además de otras utilidades. Su documentación se puede encontrar en: https://pyod.readthedocs.io/

4.2. Librerías para aprendizaje no supervisado en R

Dentro del repositorio de paquetes de R (CRAN), podemos acceder a un amplio listado de paquetes útiles para abordar la tarea de agrupamiento: https://cran.r-project.org/web/views/Cluster.html. A continuación, se enumeran los paquetes más relevantes:

- ***Stats*** es un paquete de propósito general dirigido al análisis estadístico. Como parte de sus funciones, incluye los métodos más populares para realizar agrupamiento.
- ***Cluster*** es un paquete especializado en métodos de agrupamiento que ha lanzado su versión 2.1.2 en abril de 2021.
- ***ClusterR*** es otro paquete especializado en agrupamiento que incluye métodos complementarios.

De forma similar, existe un listado amplio de paquetes R para la tarea de detección de anomalías: https://github.com/pridiltal/ctv-AnomalyDetection. ***Anomaly*** es un paquete orientado a la detección de anomalías en series temporales. Actualmente se puede descargar la versión 4.0.2, lanzada en octubre de 2021.

4.3. Otras herramientas para aprendizaje no supervisado

De forma complementaria a los paquetes y librerías anteriores, los métodos de aprendizaje no supervisado más conocidos pueden encontrarse también en herramientas basadas en otros lenguajes de programación. No obstante, es habitual que el número y variedad de técnicas sea inferior en comparación con el aprendizaje supervisado.

La herramienta ***Weka***, desarrollada en Java, incluye un conjunto reducido de métodos de agrupamiento, aunque incluye algoritmos de distintas familias. Dentro de su sección de preprocesado también pueden encontrarse algunos métodos de reducción de dimensionalidad. Weka puede descargarse como herramienta gráfica y como librería de código desde su web: https://www.cs.waikato.ac.nz/ml/weka/

KNIME permite construir soluciones de aprendizaje automático basadas en flujos de trabajos. A través de su página web (en concreto, en la sección KNIME HUB) pueden encontrarse definiciones de flujos de trabajo para tareas de agrupamiento, detección de anomalías y reducción de la dimensionalidad. Para más información, se puede consultar la documentación en su página web: https://www.knime.com/

RapidMiner también sigue la filosofía de construir desarrollos mediante flujos de trabajos. Dentro de los "operadores" que se pueden incluir en estos flujos encontramos técnicas de reducción de la dimensionalidad, detección de *outliers* y algoritmos de agrupamiento. Este último conjunto es algo más extenso que el que se puede encontrar en KNIME. Se puede descargar desde la siguiente página web: https://rapidminer.com/

Referencias

S.V. Burger. “Introduction to Machine Learning with R”. O’Reilly, 1ª edición, 212 páginas. 2018. Disponible en: https://github.com/bcaffo/regmodsbook

E. Duchesnay, T. Löfstedt, F. Younes. “Statistics and Machine Learning in Python”. Disponible en: https://duchesnay.github.io/pystatsml/

A. Géron. “Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems”. O’Reilly, 2ª edición, 483 páginas. 2019.

G. Hackeling. “Mastering Machine Learning with Scikit-Learn”. Pactkt Publishing, 221 páginas. 2014.

T. Hastie, R. Tibshirani, J. Friedman. “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”. Springer Series in Statistics, 2ª edición, 745 páginas. 2017.

G. James, D. Witten, R. Tibshirani, T. Hastie. “An Introduction to Statistical Learning with Applications in R”. Springer Texts in Statistics, 1ª edición (7ª impresión), 426 páginas. 2017. Disponible en: https://www.statlearning.com/

A. Kassambara. “Practical Guide to Cluster Analysis in R”. STHDA, 187 páginas. 2017.

M. Kubat. “An Introduction to Machine Learning”. Springer, 2ª edición, 348 páginas. 2017.

B. Lantz. “Machine Learning with R”. Pactkt Publishing, 1ª edición, 375 páginas. 2013.

A.C. Müller, S. Guido. “Introduction to Machine Learning with Python: A Guide for Data Scientists”. O’Reilly, 1ª edición (3ª impresión), 378 páginas. 2017.

A.A. Patel. “Hands-On Unsupervised Learning Using Python”. O’Reilly, 1ª edición, 400 páginas. 2019.

S. Russell, P. Norvig. “Artificial Intelligence: A Modern Approach”. Prentice Hall, 4ª edición, 1136 páginas. 2019.

S. Shalev-Shwartz, S. Ben-David. “Understanding Machine Learning: From Theory to Algorithms”. Cambridge University Press, 1ª edición, 449 páginas. 2014.