UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

Clasificadores multi-label en Scikit Learn

Presentado por Eduardo Tubilleja Calvo
en Universidad de Burgos — 28 de diciembre de 2017
Tutor: Dr. Álvar Arnaiz González
y Dr. Juan José Rodríguez Díez

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Álvar Arnaiz González y D. Juan José Rodríguez Díez, profesores del departamento de nombre departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Eduardo Tubilleja Calvo, con DNI 71298897R, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Çlasificadores multi-label en Scikit Learn".

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 28 de diciembre de 2017

V°. B°. del Tutor:

D. Álvar Arnaiz González

V°. B°. del Tutor:

D. Juan José Rodríguez Díez

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

Este proyecto trata sobre implementar diversos algoritmos de combinación clasificadores(ensembles)para multi-label sobre la librería Scikit Learn de Python, son clasificadores que combinan predicciones de otros clasificadores. Los clasificadores combinados en un ensemble son conocidos como clasificadores base. El aprendizaje automático también conocido como aprendizaje de máquinas es del subcampo de las ciencias de la computación y de la rama de inteligencia artificial, el objetivo es desarrollar unas técnicas que permitan que las máquinas aprendan. Consiste en generalizar comportamientos a partir de unos datos suministrados. El aprendizaje automático se basa en en análisis de datos y en el estudio de la complejidad computacional de los problemas.Tiene una gran variedad de aplicaciones, como pueden ser motores de búsqueda o análisis del mercado de valores entre otras.

La precisión global de los ensembles necesita que los clasificadores base predigan correctamente la clase de las mismas instancias. Tienen que ser diferentes para complementarse entre ellos. ¿Cómo puede un ensemble de clasificadores base que han sido generados por el mismo algoritmo tener distintas salidas, si las entradas son las mismas? Esto se ha conseguido en ensembles que usan distintas estrategias, los ensembles se suelen basar en la modificación del conjunto de datos de entrenamiento de clasificadores base.

En el aprendizaje supervisado, normalmente los conjuntos de datos suelen tenner solo una variable a predecir, llamado single-label. Pero desde hace un tiempo apareció el multi-label, este hace referencia a los conjuntos de datos en los que cada elemento de la base de datos puede pertenecer a la vez a más de una clase. Como por ejemplo en el ámbito del etiquetado de imágenes: en el que una imagen puede ser a la vez etiqueta "árbol", "montañaz "mar".

1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.
- **Objetivos del proyecto:** Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.
- **Conceptos teóricos:** breve explicación de los conceptos teóricos clave para la comprensión de la solución propuesta.
- **Técnicas y herramientas:** Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto.
- **Aspectos relevantes del desarrollo:** Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto.
- **Trabajos relacionados:** Este apartado sería parecido a un estado del arte de una tesis o tesina.
- **Conclusiones y líneas de trabajo futuras:** Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Junto a la memoria se proporcionan los siguientes anexos:

- **Plan del proyecto software:** En esta fase se estima el trabajo, el tiempo y el dinero que va a suponer la realización del proyecto.
- **Especificación de requisitos:** Define el comportamiento del sistema desarrollado. Sirve como documento contractual entre el cliente y el equipo de desarrollo y como documentación correspondiente al análisis a la aplicación.
- **Especificación de diseño:** Define los datos que va a manejar la aplicación, su arquitectura, el diseño de sus interfaces, sus detalles procedimentales, etc.
- **Manual del programador:** Incluye la instalación del entorno de desarrollo, la estructura de la aplicación, su compilación, la configuración de los diferentes servicios de integración utilizados o las baterías de test realizadas.
- **Manual de usuario:** Guía de usuario para el correcto manejo de la aplicación.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

2.1. Objetivos

A continuación se mostrara el esquema con todos los puntos a tratar en este proyecto.

- Implementar un algoritmo en Scikit-learn:
  - Crear los métodos fit para entrenar y predict para predecir.
  - Crear el método predict-proba para predecir probabilidades.
  - Probar que la clase funciona correctamente.
- Crear un notebook para mostrar:
  - Llamar a la clase para mostrar los resultados en notebooks de jupyter.
  - Crear una semilla para pasar a la clase.
  - Mostrar en un árbol la clasificación del conjunto.

Conceptos teóricos

3.1. Minería de Datos

Es un conjunto de reglas mediante las cuales se analizan grandes volúmenes de datos, la finalidad de esto es descubrir unos patrones, una similitud o una propensión que expliquen el comportamiento de los datos. Para hacer esto utiliza los métodos de la inteligencia artificial, estadística y redes neuronales. El objetivo del proceso de minería de datos consiste en extraer información de un conjunto de datos, luego se interpreta esta información para un uso posterior [12]. En general el proceso de la minería de datos consta de 4 etapas[9]:

Determinación de los objetivos

Se tratan los objetivos que quiere conseguir el cliente bajo un asesor especialista en minería de datos.

Preprocesamiento de los datos

Es la etapa que más tiempo se tarda en realizar el el proceso. Se seleccionan, limpian, enriquecen, reducen y transforman las bases de datos.

Determinación del modelo

Se lleva a cabo un estudio estadístico de los datos, más tarde se hace una visualización gráfica para una primera aproximación. Según los objetivos que se habían propuesto se pueden usar diferentes algoritmos de la Inteligencia Artificial.

Análisis de los resultados

Se comprueban si los datos obtenidos tienen coherencia, después se comparan con los obtenidos en los estudios estadísticos y la visualización gráfica. El

cliente es el que ve si los datos le aportan nuevo conocimiento que le permita considerar sus decisiones.

3.2. Multi-Label

La clasificación multi-label es una técnica de minería de datos, nos permite que de un conjunto de instancias de entrenamiento, podamos determinar a partir de unos atributos esenciales de dichas instancias para crear unas reglas que posteriormente se usarán para clasificar nuevas instancias [6].

3.3. Ensemble

Los métodos de ensembles combinan las predicciones de unos estimadores base, que están construidos mediante un algoritmo de aprendizaje para mejorar la solidez de un solo estimador [8]. Se distinguen dos clases de métodos de ensembles:

- En los métodos de promedio, se construyen varios estimadores de forma independiente y luego se calcula su promedio para las predicciones. En general el estimador combinado es mejor que un estimador de base única.
- En los métodos de impulso, se construyen los estimadores secuencialmente y se trata de reducir el sesgo del estimador combinado. El objetivo es combinar varios modelos débiles para conseguir un conjunto fuerte.

3.4. Disturbing Neighbors

Está organizado de la siguiente manera. La subsección 1 se explica la introducción. La subsección 2 describe el método del vecino molestón. La subsección 3 analiza experimentalmente nuestro método aplicado a los conjuntos representativos del estado del arte de SVM. La subsección 4 concluye.

Introducción

Un ensemble es un esquema de combinación de predicciones individuales llamados clasificadores base. El éxito de un ensemble requiere tanto exactitud como diversidad de sus clasificadores base. La diversidad representa como diferente son las predicciones de los clasificadores base. Si los clasificadores base siempre están de acuerdo podría no haber diferencia entre usar sólo un clasificador base o varios combinado por un método de ensemble. Entonces el poder de usar un conjunto de clasificadores base consiste en la posibilidad que algunos de ellos pueden corregir una predicción incorrecta de otros.

Es normal obtener estos clasificadores base en un ensemble usando el mismo algoritmo, así que en esta situación el proceso de entrenamiento realizado por el ensemble es la principal fuente de diversidad. La diversidad de Bagging proviene de elegir al azar diferentes instancias para entrenar a cada clasificador de base. El método Random Subspaces elige diferentes subconjuntos de atributos para entrenar a cada clasificador base. Boosting entrena de forma iterativa el conjunto de clasificadores base, modificando los pesos de las instancias para entrenar al clasificador actual. Estos nuevos pesos se calculan a partir del error de entrenamiento en el clasificador base anterior, por lo que cada nuevo clasificador de base llega más especializado en instancias que han sido mal clasificados antes. A veces los clasificadores base son muy estables y el algoritmo de entrenamiento del ensemble no es suficiente para proporcionar el nivel deseado de diversidad.

SVM (Support Vector Machine) calcula un hiperplano óptimo que separa el espacio de entrada en dos regiones correspondientes a las clases de un conjunto de datos de dos clases. Si el conjunto de datos no es separable linealmente, el hiperplano puede construirse sin tal problema en el espacio de características dado por una función kernel. Cuando no se usa kernel, se dice que el kernel es lineal. El kernel lineal es más apropiado para datasets linealmente separables. Sin embargo, también es una opción interesante para otros conjuntos de datos, ya que es el más rápido y existen implementaciones optimizadas. Además, si no es posible la separación completa, pueden introducirse variables de holgura para permitir errores de entrenamiento. Por lo tanto, el kernel lineal es una opción muy competitiva cuando se deben construir numerosos SVM, como ocurre con los ensambles. Por esa razón, este documento está enfocado solo en SVM lineal. Sin embargo, algunas de sus conclusiones podrían ser válidas para otros kernels. Se sabe que SVM es un clasificador muy estable, por lo que se espera que los conjuntos de SVM se beneficien de estrategias que contribuyan a aumentar su diversidad.

Los Disturbing Neighbors (DN) o vecinos molestones se han utilizado con éxito para mejorar la diversidad en los bosques. DN usa un clasificador de 1-Nearest Neighbour (1-NN) para construir un conjunto de características adicionales que se agregan al conjunto de datos de entrenamiento de cada clasificador base. Este clasificador 1-NN es diferente para cada clasificador base. Las características compiladas son la predicción 1-NN más un conjunto booleano de características que indican cuál es el vecino más cercano. El conjunto de datos de entrenamiento original se transforma en un conjunto de datos aumentados, que es diferente para cada clasificador base, independientemente del esquema de conjunto en el que se va a utilizar.

A diferencia de SVM, los árboles de decisión son muy sensibles a pequeños cambios en el conjunto de datos de capacitación. La motivación de este documento es probar DN dentro de conjuntos SVM para ver si la precisión y

la diversidad aumentan, al igual que en los bosques, a pesar de la estabilidad SVM.

Método

El método DN trabaja en cada clasificador base de la siguiente manera:

1. m instancias son seleccionadas aleatoriamente de el conjunto de datos de entrenamiento para construir un clasificador 1-NN. El valor m usa valores muy pequeños.
2. Dimensiones usadas para calcular distancia euclidea en el clasificador 1-NN son también seleccionadas aleatoriamente. Al menos el 50 % de los atributos son seleccionados.
3. Luego m+1 nuevas características son añadidas al conjunto de entrenamiento. Una de las características adicionales es la clase predecida por el clasificador 1-NN para cada instancia x, y la otra m son características booleanas, todos los conjuntos falsos excepto uno corresponden al vecino más cercano para esa instancia.
4. El clasificador base está entrenado usando las características originales mas las nuevas características de m+1.

Por lo tanto, el proceso normal de entrenamiento de los clasificadores básicos se altera añadiendo estas nuevas características del clasificador 1-NN. Es por eso que el método se llama vecinos molestones. La aleatoriedad aumenta la diversidad y se debe a:

- Los vecinos utilizados en cada clasificador 1-NN se seleccionan aleatoriamente. Por lo tanto, sus predicciones y las características booleanas son diferentes para cada clasificador base.
- Las dimensiones utilizadas para calcular las distancias euclidianas también se eligen de forma aleatoria, así que si dos clasificadores básicos tienen al menos los mismos m vecinos, las predicciones 1-NN y las características booleanas podrían ser diferentes.

El valor m es muy pequeño ya que el objetivo no es crear un clasificador 1-NN muy preciso, sino uno diferente cada vez. Las características m booleanas pueden ser tenidas en cuenta por el SVM resultante, por lo que se espera que cada SVM sea diferente. Como las predicciones de clase 1-NN son nominales, deben transformarse en binarias características para ser calculadas por el algoritmo de entrenamiento SVM. Estas características binarias también dividen el espacio de entrada en regiones cuyo número es igual o menor que el número de

clases. En realidad, cada región resultante de esta nueva división es la unión de las regiones de Voronoi correspondientes a los vecinos que comparten la misma clase. Así que otra vez, tenemos un conjunto de atributos booleanos que pueden cambiar los coeficientes de SVM. Por lo tanto, las dimensiones adjuntas por DN pueden ser utilizadas por cada SVM en el ensemble. La aleatoriedad introducida hace que estas dimensiones sean diferentes para cada SVM, por lo que se obtienen diversos hiperplanos cada vez.

Resultados

Conclusión

Disturbing Neighbors es un método para alterar el proceso de entrenamiento normal de los clasificadores base en un ensemble, mejorando su diversidad y mejorando la precisión general del ensemble. Disturbing Neighbors crea nuevas características utilizando un clasificador 1-NN. Estas características son la salida 1-NN más un conjunto de atributos booleanos que indican cuál es el vecino más cercano. El clasificador 1-NN se crea utilizando un pequeño subconjunto de instancias de entrenamiento seleccionadas al azar del conjunto de datos original. Las dimensiones utilizadas para calcular la distancia euclidiana también se seleccionan de forma aleatoria. Estas dos fuentes de aleatoriedad son las razones por las que las características creadas son diferentes cada vez, por lo que cuando estas nuevas características se usan para entrenar clasificadores base, la diversidad aumenta. La estadística de Kappa y los diagramas de movimiento de Kappa muestran que DN proporciona una diversidad adicional a los alumnos de base de SVM. La validación experimental también muestra que DN-SVM no mejora significativamente la precisión de SVM, mientras que los conjuntos de SVM que usan DN son significativamente mejores que las versiones sin DN. De modo que esta mejora solo puede venir del incremento en diversidad obtenido al aplicar DN.

3.5. Referencias

Las referencias se incluyen en el texto usando cite [10]. Para citar webs, artículos o libros [3].

3.6. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.1: Autómata para una expresión vacía

3.7. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.8. Tablas

Igualmente se pueden usar los comandos específicos de LATEXo bien usar alguno de los comandos de la plantilla.

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

4.1. GitHub:

Es una plataforma para alojar proyectos utiliza git como sistema de control de versiones. Se organiza por tareas(milestones e issues). Utiliza el framework Ruby on Rails[1].

Podemos acceder a él a través del siguiente enlace: https://github.com/

Ventajas:

- Es uno de los repositorios mas usados, por lo que es fácil encontrar información en Internet para resolver cualquier duda.
- El código es público por lo que cualquiera puede proponer cambios en el mismo, seguirte y ver el proyecto.
- Las distintas versiones del código están alojadas en la nube por lo que si perdemos el contenido de nuestro ordenador, podremos recuperarlo.

Desventajas:

- También puedes tener proyector privados pero para ello tienes que utilizar una cuenta de pago.

4.2. Spyder:

Es un entorno de desarrollo interactivo para el lenguaje de Python, es de código abierto. Tiene funciones avanzadas de edición, pruebas interactivas, depuración e introspección. También es un entorno informático numérico y tiene diversas bibliotecas que podemos utilizar, como pueden ser numpy [5].

Podemos acceder a él a través del siguiente enlace: http://pythonhosted.org/spyder/

4.3. LaTeX:

Se usa para la creación de documentos que necesiten una alta calidad tipográfica, como puede ser en artículos o libros científicos [4].

Ventajas:

- Es software libre, por lo que no requiere ningún coste.
- No te tienes preocupar por el diseño, ya que la herramienta se encarga de ello.

Desventajas:

- Si eres principiante necesitas un tiempo de aprendizaje para saber como funciona.

Podemos acceder a él a través del siguiente enlace: https://www.latex-project.org/

4.4. Jupyter Notebook:

Es una aplicación web de código abierto, con él podemos crear y compartir documentos, que nos permiten visualizar los resultados al ejecutar nuestro código, ya sean imágenes, árboles...que otros entornos de desarrollo (como Spyder mencionado anteriormente) no nos permiten esto. Soporta más lenguajes pero nosotros lo usaremos para el lenguaje de Python [2].

Podemos acceder a él a través del siguiente enlace: http://jupyter.org/

4.5. Scikit-learn:

Es una librería de aprendizaje de software libre en el lenguaje de programación de Python. Es una herramienta de las más utilizadas para la minería de datos y el análisis de datos[11]. Esta basada en el aprendizaje automático, para ello se consideran un conjunto de n muestras y se intenta predecir las propiedades de los datos desconocidos. Podemos separar los problemas de aprendizaje en dos[7]:

- Aprendizaje supervisado: En el que los datos vienen con atributos adicionales que queremos predecir. Dos de las tareas mas comunes son la clasificación y la regresión. En las de clasificación el programa debe aprender a predecir en que categoría o clase irán los nuevos datos según las nuevas observaciones, como sería predecir si el precio de una acción bajara o subirá. En lo de regresión el programa debe predecir el valor de una variable de respuesta continua, como sería predecir las ventas de un nuevo producto.
- Aprendizaje no supervisado: Consiste en agrupar observaciones relaciones,dentro de los datos del entrenamiento. Clustering es la que más se utiliza para explorar un conjunto de datos.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] genbetadev. Github, 2011. [Online].

[2] jupyter. Jupyter-notebook, 2017. [Online].

[3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[4] latex project. Latex, 2017. [Online].

[5] pythonhosted. Spyder, 2017. [Online].

[6] researchgate. Multi-label, 2014. [Online].

[7] scikit learn. Documentación scikit-learn, 2017. [Online].

[8] scikit learn. Ensemble, 2017. [Online].

[9] sinnexus. Minería de datos, 2016. [Online].

[10] Wikipedia. Latex — wikipedia, La enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].

[11] Wikipedia. Documentación scikit-learn, 2017. [Online].

[12] Wikipedia. Minería de datos, 2017. [Online].