

Algoritmo ID3 mixto (quórum e incertidumbre)

Inteligencia Artificial (IS) 2018/19 – Propuesta de trabajo

Agustín Riscos Núñez

1. Introducción y objetivo

El algoritmo ID3 es un ejemplo de aprendizaje supervisado, donde se construye un árbol de decisión consistente con un conjunto de ejemplos de entrenamiento. A medida que se van añadiendo nodos al árbol, los ejemplos se van repartiendo entre las ramas (en función del valor del atributo que ha sido elegido como nodo), pudiendo darse el caso de que alguna rama incluso se quede sin ningún ejemplo.

El **objetivo principal** de esta propuesta es implementar una versión modificada del algoritmo ID3 que incluya un parámetro inicial que imponga un “quórum” (es decir, una cantidad mínima de ejemplos) que queramos imponer para que las ramas del árbol sean “fiables”. En los casos en que tras una ramificación el número de ejemplos restantes sea inferior al quórum, entonces en lugar de continuar con la ejecución del algoritmo ID3, se añadirá al árbol en esa rama un nodo hoja que invoque una llamada al algoritmo Naive Bayes.

El árbol mixto así construido tendrá entonces tres tipos de nodos:

1. nodos interiores (cada uno con un atributo asociado),
2. nodos hoja-categoría (cada “categoría” será uno de los valores del atributo objetivo que estamos usando para el problema de clasificación),
3. nodos hoja-truncada (sin categoría asociada, esto será simplemente un puntero que lance una ejecución del algoritmo Naive Bayes para el ejemplo que esté siendo clasificado).

Esto evidentemente generará dos tipos de salida cuando se utilice el árbol para clasificar nuevos ejemplos: o bien devolverá un valor de clasificación sin incertidumbre (en el caso de que al nuevo ejemplo le haya correspondido una rama “normal” o “clásica”) o bien devolverá un valor de clasificación con una probabilidad asociada (en el caso de que la rama fuese truncada por no haberse alcanzado el quórum, tal y como se ha explicado en el párrafo anterior).

Para ello será necesario alcanzar los siguientes objetivos **específicos**:

1. Idear una representación en Python de un árbol mixto.
2. Implementar una función de clasificación adaptada, que dado un árbol mixto y un ejemplo nuevo, devuelva el valor de clasificación correspondiente, junto con la probabilidad asociada si se llega a hacer la llamada a Naive Bayes.
3. Implementar una versión modificada del algoritmo ID3, de acuerdo con lo explicado anteriormente.
4. Implementar una medida de rendimiento adaptada a los árboles mixtos generados por el algoritmo modificado.
5. Realizar pruebas con conjuntos de entrenamiento extraídos de algún repositorio online, o bien preparados por los propios alumnos.
6. Documentar el trabajo en un fichero con formato de artículo científico, explicando con precisión las decisiones de diseño en la representación del problema, los resultados obtenidos en las pruebas de evaluación (prestando especial atención a cambios de rendimiento relacionados con cambios de valor inicial de quórum).

7. Realizar una presentación de los resultados obtenidos en la defensa del trabajo.

Para que el trabajo pueda ser evaluado, se deben satisfacer TODOS los objetivos específicos al completo: el trabajo debe ser original, estar correctamente implementado y funcionar perfectamente, los experimentos se deben haber llevado a cabo y analizados razonadamente, el documento debe ser completo y contener entre 8 y 10 páginas (no más, no menos), y se debe realizar la defensa con una presentación de los resultados obtenidos.

2. Descripción del trabajo

A continuación se introduce la metodología a seguir para el correcto desarrollo del trabajo.

2.1. *Implementación de los algoritmos*

La estructura de datos elegida para la representación en Python de los árboles mixtos será preferiblemente original, pero podrá estar basada en estructuras similares utilizadas en librerías de Python que trabajen con árboles de decisión. Análogamente para la estructura de datos que se utilice para los ejemplos del conjunto de entrenamiento.

Los objetivos específicos 2, 3 y 4 requieren desarrollo de código original que o bien recibe como entrada o bien genera como salida árboles mixtos. Para la aplicación del algoritmo Naive Bayes, se tendrá más flexibilidad, permitiendo tanto desarrollos originales como utilización de librerías existentes (o adaptación de las mismas al formato elegido para el trabajo).

El trabajo no debe diseñarse de forma específica para unos datos concretos, sino que debe permitir realizar experimentos con otros conjuntos de entrenamiento diferentes (distintos tamaños, otros atributos, etc).

El conjunto de ejemplos se proporcionará a través de un fichero aparte, que puede ser csv, arff, o tener las restricciones de formato que se deseen, siempre que se aporten las indicaciones necesarias en un fichero README.txt (ver apartado 3).

2.2. *Experimentación*

La experimentación se debería plantear como una batería de pruebas con distintos valores de quórum, comparando los rendimientos obtenidos en cada caso con el árbol obtenido por el algoritmo ID3 sin modificaciones. Para que tenga sentido aplicar la versión modificada del algoritmo, deberá comprobarse que para el ejemplo elegido aparecen en el árbol devuelto por el ID3 clásico varios nodos a los que les corresponden pocos ejemplos (o incluso ninguno).

2.3. Documentación

En el fichero `plantilla-trabajo.doc` se muestra una sugerencia de estructura y formato de estilo artículo científico correspondiente a la documentación del trabajo. Este formato es el del *IEEE conference proceedings*, cuyo sitio web *guía para autores* [1] ofrece información más detallada y plantillas para Word y LaTeX.

El artículo deberá tener una extensión **entre 8 y 10 páginas**, y la estructura general del documento debe ser como sigue: en primer lugar realizar una **introducción** al trabajo explicando el objetivo fundamental, incluyendo un breve repaso de antecedentes en relación con la temática del trabajo y con los métodos empleados (mencionar referencias bibliográficas), a continuación describir la **estructura** del trabajo, las **decisiones de diseño** que se hayan tomado a lo largo de la elaboración del mismo, y la **metodología** seguida al implementarlo (nunca poner código, pero sí pseudocódigo), y seguidamente detallar los **experimentos** llevados a cabo, **analizando los resultados** obtenidos. Por último, el documento debe incluir una sección de **conclusiones**, y una **bibliografía** donde aparezcan no sólo las referencias citadas en la sección de introducción, sino cualquier documento consultado durante la realización del trabajo (incluidas las referencias web a páginas o repositorios).

2.4. Mejoras

Aunque no sea obligatorio, sí se tendrá en cuenta en la calificación la incorporación de un interfaz o menú amigable que facilite la experimentación o la presentación de resultados. También podrán recibir puntuación adicional otras mejoras o añadidos que se incorporen al trabajo más allá de los requisitos mínimos que se mencionan en la lista de objetivos específicos.

Por ejemplo: probar opciones alternativas a Naive Bayes cuando no se alcanza el quórum, diseñar e implementar un algoritmo de poda sobre árboles mixtos, etc

2.5. Presentación y defensa

El día de la defensa se deberá realizar una pequeña presentación (PDF, PowerPoint o similar) de 10 minutos en la que participarán activamente todos los miembros del grupo que ha desarrollado el trabajo. Esta presentación seguirá a grandes rasgos la misma estructura que el documento, pero se deberá hacer especial mención a los resultados obtenidos y al análisis crítico de los mismos. Se podrá usar un portátil (personal del alumno), diapositivas y/o pizarra. En los siguientes 10 minutos de la defensa, el profesor procederá a realizar preguntas sobre el trabajo, que podrán ser tanto del documento como del código fuente.

3. Criterios de evaluación

Para que el trabajo pueda ser evaluado, se deberá satisfacer los objetivos concretos descritos en el apartado 1 (todos y cada uno de ellos, si no, el trabajo obtendrá una nota de suspenso). Uno de los alumnos del equipo deberá subir a través del formulario disponible en la página de la asignatura un fichero comprimido .zip, que contenga:

- **Una carpeta con el código fuente.** Dentro de dicha carpeta tiene que haber un fichero `README.txt`, que resuma la estructura del código fuente, e indique cómo usar la interfaz (si se ha implementado), o al menos cómo hacer pruebas con las funciones implementados, incluyendo ejemplos de uso. Asimismo se deberá indicar cómo reproducir los experimentos realizados. Es importante la coherencia de este fichero con la defensa.
- **El documento – artículo en formato PDF.** Deberá tener una extensión mínima de 8 páginas, y máxima de 10. Deberá incluir toda la bibliografía consultada (libros, artículos,

technical reports, páginas web, códigos fuente, diapositivas, etc.) en el apartado de referencias, y mencionarlas a lo largo del documento.

Para la evaluación se tendrá en cuenta el siguiente criterio de valoración, considerando una nota máxima de 3 en total para el trabajo:

- **El código fuente (1 punto):** se valorará la claridad y buen estilo de programación, corrección y eficiencia de la implementación, y calidad de los comentarios. La claridad del fichero README.txt también se valorará. En ningún caso se evaluará un trabajo con código copiado directamente de internet o de otros compañeros.
En este trabajo está permitido utilizar en parte librerías de Python existentes, aunque en este apartado se valorará exclusivamente el código original desarrollado por los alumnos.
- **Experimentación (0.5 puntos):** se valorará la cantidad de experimentos realizados, la calidad de los datos utilizados (dimensión de los conjuntos de entrenamiento utilizados, comprobaciones del número de ramas afectadas por la modificación del algoritmo, validación cruzada, etc), y la claridad con la que se incluyan los datos obtenidos en la memoria (tablas y/o gráficas).
- **El documento – artículo científico (1 punto):**
 - Se valorará el uso adecuado del lenguaje y el estilo general del documento (por ejemplo, el uso de la plantilla sugerida).
 - Se valorará en general la claridad de las explicaciones, el razonamiento de las decisiones, y especialmente el análisis y presentación de resultados en las secciones de experimentación y conclusiones.
 - Igualmente, no se evaluará el trabajo si se detecta cualquier copia del contenido del documento. La sección de referencias deberá incluir menciones a todas las pertinentes fuentes consultadas.
- **La presentación y defensa (0,5 puntos):** se valorará la claridad de la presentación y la buena explicación de los contenidos del trabajo, así como, especialmente, las respuestas a las preguntas realizadas por el profesor.
- **Mejoras:** Se valorarán hasta con 0.5 puntos extra sin superar el máximo de 3 puntos totales del trabajo.

IMPORTANTE: Cualquier **plagio, compartición de código** o uso de material que no sea original y del que no se cite convenientemente la fuente, significará automáticamente la **calificación de cero** en la asignatura para **todos los alumnos involucrados**. Por tanto, a estos alumnos **no se les conserva**, ni para la actual ni para futuras convocatorias, **ninguna nota** que hubiesen obtenido hasta el momento. Todo ello sin perjuicio de las correspondientes **medidas disciplinarias** que se pudieran tomar.

4. Referencias

[1] Plantilla IEEE. https://www.ieee.org/conferences_events/conferences/publishing/templates.html