Inteligencia Artificial
Artículo de Investigación

Pablo Alvarado
Estudiante
Universidad Rafael Landívar
Carné: 1104017
pablo.andre_alvarado@hotmail.com

La Inteligencia Artificial es una rama de las ciencias de la computación que a su vez se subdivide en una variedad de campos de estudio enfocados cada uno en distintas áreas de análisis que buscan responder a diferentes preguntas como ¿quién?, ¿qué?, ¿qué harán?, ¿cómo?, ¿por qué?. Para responder a estas preguntas es necesario contar con grandes cantidades de información, las cuales sirven principalmente para poder agruparla e inferir comportamientos, así como poder visualizar tendencias y poder llegar a realizar predicciones sobre temas en específico. Cabe resaltar que en el área de Inteligencia Artificial se utiliza mucho de la ayuda de la probabilidad y la estadística, ya que son de gran utilidad para predecir comportamientos o realizar inferencia sobre información previamente existente, que a su vez puede ser clasificada a conveniencia para sacarle el mayor provecho posible. Cabe resaltar que hoy en día, los algoritmos utilizados por la rama de la Inteligencia Artificial siguen siendo los mismos que se desarrollaron hace décadas al inicio de esta rama de las ciencias de la computación. El motivo por el cuál a día de hoy se pueden utilizar sacándoles mucho más provecho es que hoy en día existen grandes cantidades de información con lo cuál los algoritmos son mucho más eficientes.

1. Introducción

El curso de Inteligencia Artificial impartido en la Universidad Rafael Landívar se enfoca principalmente en distintos métodos de búsqueda de información e inferencia a partir de la misma, así como algoritmos utilizados en videojuegos clásicos en sus inicios. También se estudió la inferencia en grafos a partir de varios factores con lo cual se obtenía la mejor ruta. Finalmente se estudió el análisis de probabilidades conjuntas y condicionales mediante tablas de distribución y principalmente mediante el Teorema de Bayes y sus variantes. El proyecto que se describirá a lo largo de este artículo tiene como objetivo crear un recomendador de películas a partir de una colección de datos considerablemente pequeña de un ranking publicado por Kagle de las mejores 5000 películas votadas en IMDB. Debido a que a pesar de la data inicial no se conoce precedencia de orientaciones dadas por gustos del usuario se debe de utilizar algún método de arranque en frío para recopilar información y brindar mejores recomendaciones conforme se utilice la aplicación.

2. Estado del Arte

Machine learning es una forma de inteligencia artificial especializada en proveer un conjunto de métodos que pueden automáticamente detectar patrones en los datos y utilizar los patrones descubiertos para predecir datos futuros o tomar decisiones en un entorno de incertidumbre. Algunas fuentes se refieren a machine learning a los métodos computacionales que utilizan experiencia para mejorar su rendimiento o realizar predicciones con cierto nivel de precisión. Por experiencia se entiende al conjunto de de datos utilizado para el análisis y aprendizaje. Algunas medidas de calidad de los algoritmos utilizados en machine learning son la complejidad temporal y espacial. Para estudiar la eficiencia de un algoritmo se suele analizar la eficiencia de la ejecución a medida que el tamaño de datos de entrada vaya aumentando. Existe una amplia variedad de problemas que se busca resolver con machine learning, entre los cuales podemos encontrar:

- Clasificación: Asignar una categoría a cada elemento del conjunto estudiado. Como por ejemplo si un elemento (correo electrónico) es spam o es jam.
- Regresión: Predecir un valor real a cada elemento. Como por ejemplo, predecir el costo mensual del uso de un servicio en la nube.
- Ranking: Ordenar elementos de acuerdo a un criterio. Como por ejemplo, ordenar páginas web más relevantes.
- Culstering: Partición de elementos en regiones, cada una compuesta por elementos semejantes. Como por ejemplo, identificar grupos de usuarios en las redes sociales.
- Aprendizaje Múltiple: Transformar una representación inicial en una representación de menor dimensión que preserva algunas propiedades, como por ejemplo la comprensión de imágenes digitales.

2.1. Clasificador de Patrones con Bayes

Una red bayesiana, es un grafo acíclico en el que cada nodo representa una variable y cada arco una dependencia probabilística, son utilizadas para:

- Proveer una forma compacta de representar el conoci-

miento.

- Proveer métodos flexibles de razonamiento.

El obtener una red bayesiana a partir de datos, es un proceso de aprendizaje que se divide en dos etapas, el aprendizaje estructural y el aprendizaje paramétrico.

Utilizar el teorema de Bayes en cualquier problema de aprendizaje automático conlleva el beneficio de que es posible estimar las prioridades de cualquier hipótesis consistente con el conjunto de datos de entrenamiento para así poder escoger la hipótesis más probable. Para estimar estas probabilidades se han destacado numerosos algoritmos, entre los que cabe destacar el algoritmo Naive Bayes.

2.2. Aprendizaje Automático

El aprendizaje es definido como Çualquier proceso a través del cual un sistema mejora su eficiencia"[Felgaer, P. et al, 2003]. Una de las características centrales de los sistemas inteligentes es aprender. Un aspecto importante en el aprendizaje inductivo es el obtener un modelo que represente el dominio de conocimiento y que sea accesible para el usuario. Resulta importante obtener la información de dependencia entre las variables involucradas en el fenómeno, en los sistemas donde se desea predecir el comportamiento de algunas variables desconocidas basados en otras condiciones; una representación del conocimiento que es capaz de capturar esta información sobre las dependencias entre las variables son las redes bayesianas [Ramoni Sebastiani, 1999].

2.3. Naive Bayes

El clasificador Naive Bayes aparece por primera vez en la literatura de aprendizaje automático a finales de los años ochenta, con el objetivo de comparar su capacidad predictiva con la de métodos más sofisticados. De manera gradual los investigadores de esta comunidad de aprendizaje automático se han dado cuenta de su potencialidad y robustez en problemas de clasificación supervisada.

Dado un ejemplo x representado por k valores, el clasificador Naive Bayes se basa en encontrar la hipótesis más probable que describa a ese ejemplo en específico. Si la descripción de ese ejemplo viene dada por los valores

$$<a_1, a_2, ..., a_n>$$

La hipótesis más probable será aquella que cumpla la probabilidad de que conocidos los valores que describen a este ejemplo, éste pertenezcan a la clase vj (donde vj es el valor de la función de clasificación f(x) en el conjunto finito V).

Por el teorema de Bayes podemos estimar P(vj) contando las veces que aparece el ejemplo vj en el conjunto de entrenamiento y dividiéndolo por el numero total de elementos que forman el conjunto.

2.4. Ejemplo Naive Bayes

Un clásico ejemplo del Algoritmo de Naive Bayes es calcular a partir de un grupo de palabras previamente clasificadas la probabilidad de que un correo electrónico que contenga palabra o un mensaje determinado sea clasificado como spam o como jam (no spam). Para la realización del siguiente ejemplo se toma como base la siguiente tabla con las probabilidades de que cada palabra sea considerado como spam o como jam.

Word	P(w|spam)	P(w|ham)	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gary	0.00002	0.00021	-11.8	-8.9
would	0.00069	0.00084	-19.1	-16.0
you	0.00881	0.00304	-23.8	-21.8
like	0.00086	0.00083	-30.9	-28.9
to	0.01517	0.01339	-35.1	-33.2
lose	0.00008	0.00002	-44.5	-44.0
weight	0.00016	0.00002	-53.3	-55.0
while	0.00027	0.00027	-61.5	-63.2
you	0.00881	0.00304	-66.2	-69.0
sleep	0.00006	0.00001	-76.0	-80.5

P(spam | w) = 98.9

Figura 1. Tabla de probabilidades. Ejemplo Mensajes Spam y Jam

Tal y como se puede observar en la Figura 1, en la parte superior de la tabla se encuentra la probabilidad de que un mensaje en general sea Spam o se Jam, también existe un universo de palabras, las cuales conforman el siguiente mensaje de correo electrónico: *Gary would you like to lose weight while you sleep.* Claramente este es un mensaje Spam, sin embargo la forma en que una inteligencia artificial puede deducir una respuesta acertada al problema debe de recurrir a información histórica que debe estar normalizada y clasificada. En este caso se cuenta con la probabilidad de que la palabra en específico sea Spam o Ham.

Uno de los problemas con los que se enfrenta el algoritmo Naive Bayes es que las probabilidades de que se manejan en los elementos individuales que conforman un evento suelen ser muy pequeñas por lo que al analizar un evento que contenga una cantidad de elementos relativamente grande, una computadora pierde los decimales y por tanto las aproximaciones pierden exactitud. Una forma de solucionar este problema es utilizar leyes de logaritmos. En lugar de multiplicar las probabilidades se suman sus equivalentes logarítmicos con lo cual se evita perder decimales. Finalmente para obtener las probabilidades de que cierto evento ocurra es necesario utilizar el valor final de la suma de logaritmos, tanto positivo como negativo (ocurra o no el evento) u se utiliza el siguiente calculo utilizando exponenciales.

$$<e^x / e^x + e^y>$$

En donde x representa el numero logarítmico de la probabilidad positiva y y representa el número logarítmico de la probabilidad negativa. En este caso, si se utiliza la formula mostrada arriba obtendríamos el resultado de la probabilidad positiva. Para obtener el resultado de la probabilidad negativa sería necesario colocar y en el numerador o simplemente restarle a al número uno el primer resultado obtenido.

3. Solución del Problema

3.1. Normalización de la Información

Generalmente en un conglomerado de información cuenta con una variedad de categorías y no todas son necesariamente útiles. Dependiendo del contexto se puede depurar la información para analizar únicamente información que sea de utilidad.

Llevar a cabo esta acción puede mejorar considerablemente las complejidades tanto temporales como espaciales de una implementación, ya que cuando la información es relativamente grande (y lo deben ser para brindar mejores recomendaciones, aproximaciones y predicciones) la información analizada puede hacer la diferencia entre una implementación óptima y una deficiente.

En el caso del recomendador de películas, con la información brindada se utilizaron únicamente las categorías que podían brindar mayor cantidad de coincidencias y por lo tanto recomendaciones más precisas. Las categorías utilizadas como parte de información general y necesaria de la película.

Al usuario se le muestra la siguiente información: Titulo de la Película, Director de la Película, Año de lanzamiento, Clasificación del título y cuando la película haya sido votada también se mostrará la probabilidad de que al usuario le guste la película mostrada.

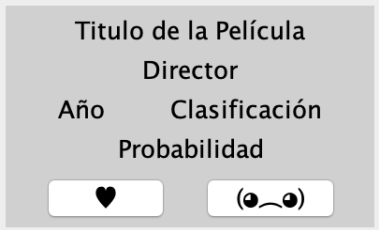

Figura 2. Información de la película mostrada al usuario

La información que se utiliza para el funcionamiento del recomendador de películas es la siguiente:

- Duración: En el caso de la duración de la película se utiliza como un rango para tener coincidencias con películas que tengan duraciones similares.
- Géneros: Cada película cuenta con un arreglo de géneros.
- Lenguaje: Se refiere al idioma en que se encuentra la película.
- País: Hace referencia al país en cual se llevó la producción de la película.
- Categoría de Contenido: Es la categoría en las que se clasifican las películas como por ejemplo para todo publico, adultos, infantil, entre otros.
- Año de Lanzamiento: El año en el cual fue estrenada la película.
- Director: La persona que dirigió la producción del título.
- Actores 1, 2 y 3: Actores principales que aparecen en la entrega.

3.2. Arranque en Frío

El arranque en frío es un problema para el ámbito computacional enfocado en sistemas que se basan en la recopilación de información para poder inferir tendencias y predicciones. En el caso principal al cual se enfoca este artículo de investigación, el arranque en frío es un problema al cuál se enfrenta el recomendador de películas desarrollado. Aunque se cuente con información suficiente para realizar recomendaciones no se cuenta con datos históricos para poder realizar sugerencias a través de las tendencias marcadas por otros usuarios.

La forma en que se arrancó el programa fue utilizando información presente en la data inicial como la cantidad de likes que tiene cada Película en FaceBook. De esta forma se ofrece al usuario una variedad de películas populares que puede que le gusten.

Asimismo la plataforma cuenta con un buscador que permite al usuario encontrar una película que coincida su búsqueda ya que de lo contrario tomaría demasiado tiempo encontrar una película en la plataforma de una manera secuencial.

3.3. Interacción del Usuario

Cuando el usuario ingresa a la plataforma puede visualizar tres carruseles, de películas.

Figura 3. Interfaz principal para interacción del usuario

El carrusel superior es un catálogo de películas que por el momento se encuentra indefinido, ya que está destinado a las búsquedas realizadas por el usuario, por lo que tomarán un valor cuando el usuario lleve a cabo la búsqueda de algún titulo.

El segundo carrusel muestra un Top de Películas en base a la cantidad de likes de cada título en la red social FaceBook. En este caso la información ya fue analizada y normalizada para su uso durante la inicialización del programa, por lo que ya se puede interactuar con esta sección.

El tercer carrusel muestra un Top de Recomendaciones personalizadas, las cuales se pueden obtener a partir de que el usuario interactúe con la plataforma y brinde su gusto o rechazo acerca de uno o varios títulos.

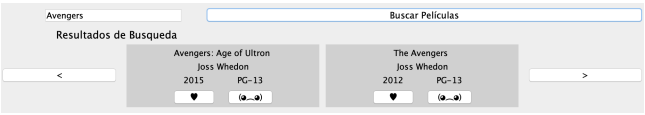

Figura 4. Buscador de películas

Cada uno de los carruseles mencionados anteriormente cuenta con una serie de películas en su interior y cada una cuenta con un botón de ”Me Gustaz con un botón de ”No Me Gusta”, representadas por un corazón y una cara triste respectivamente. Con estas acciones el usuario tiene la posibilidad de marcar una película como favorita o brindarle una calificación negativa, con lo cual se recopila la información y afecta a todas las películas que cuenten con características coincidentes.

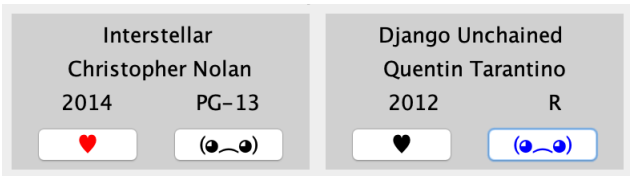

Figura 5. Acciones de me gusta y no me gusta

3.4. Solución

El recomendador se construyó a partir de los conceptos del algoritmo Naive Bayes. Como primer paso se normalizó un archivo .csv del Top 5000 de películas ofrecido por el IMDB. Al momento en que se inicializó el programa se definieron como campos clave para la recomendación de películas: la duración, color, géneros, idioma, país, clasificación, año de lanzamiento, director, actor (1), actor (2) y actor (3).

La forma en que se obtienen las probabilidades es la siguiente: Al inicio del programa se recorre la totalidad de los datos ingresados a través del archivo .csv y se obtienen los universos (apariciones) de cada aspecto en la película, por ejemplo, si una película fuera dirigida por Quentin Tarantino, la aplicación de recomendaciones busca entre todas las películas cuántas fueron dirigidas por la misma persona, obteniendo de esta forma un normalizador para dicho elemento y todos los elementos coincidentes (Todas las películas dirigidas por la misma persona). En este caso el normalizador es válido únicamente para la categoría de director, una de las muchas categorías que influyen en la recomendación final.

En el caso de parámetros numéricos como la duración de la película y el año de lanzamiento se tomaron en cuenta intervalos numéricos. Para el año de lanzamiento se tomaron como coincidencias las películas de la misma década y en el caso de la duración de la película se tomaron en cuenta intervalos de diez minutos como coincidencias, ya que no representa una variación significativa de tiempo que pudiera afectar los resultados. Asimismo al utilizar intervalos se abre la posibilidad a tener un numero mayor de coincidencias, con lo que se obtiene un mayor numero de recomendaciones que en esencia tienen características muy parecidas entre si.

Los valores de los numeradores que forman parte de la probabilidad de una categoría en específico junto con el normalizador encontrado al inicio de ejecución de la aplicación, se obtienen a partir de la interacción del usuario con la aplicación. Si el usuario vota una película como favorita, todas las categorías que conforman esa película elevan su **indicador positivo** (por categoría) en una unidad y se recorre nuevamente la colección de datos para aumentar en una unidad las únicamente categorías que sean coincidentes con la película votada. Por ejemplo, si se vota como favorita una película del genero de acción, se elevará en una unidad la categoría del genero de todas las películas que también sean del genero de acción y de esta misma manera con todas las categorías de las películas.

De la misma manera ocurre cuando se vota una película negativa, la única diferencia es que en lugar de modificar su indicador positivo se modifica su **indicador negativo**.

Una vez se hayan modificado todos las categorías de las películas se utilizan leyes de logaritmos para evitar la perdida de decimales durante la ejecución, tal y como se indicó anteriormente en este artículo. Se procede a sumar los logaritmos de cada categoría, en el caso de las categorías que conforman un voto positivo se haría de la siguiente forma:

$$<ln(cat1)+ln(cat2)+...+ln(p(LikeMovie)>$$

Que a su vez, las categorías son obtenidas de la siguiente manera:

$$<cat1 = coincidenciasPositivas/Normalizador(Universo)>$$

Sin embargo cuando existen muy pocos datos, el algoritmo se encuentra con el siguiente problema:

Cuando aún no existen suficientes votaciones, muchas categorías se encuentran como una fracción igual a cero, ya que no existe ninguna coincidencia y el denominador no aporta ninguna diferencia ante esta situación. El problema surge cuando se convierte la probabilidad por categoría (fracción mencionada anteriormente) a un número logarítmico, ya que ln(0) no existe. Tampoco se puede omitir la categoría ya que sería obviar su participación en la probabilidad de que la película pueda gustarle o no al usuario, por lo que se optó

por la siguiente solución. Se asume un numerador mínimo, es decir, la unidad. Sin embargo ya que esto contaría como una coincidencia para esa categoría, se duplica el universo (denominador) con lo cual se reduce sin importar la categoría o el ámbito (positivo o negativo) con lo cual se priorizan los votos del usuario.

Cuando se recopilan datos del usuario suficientes, se anula la configuración indicada, con lo cual, mientras mayor sea la interacción del usuario con la plataforma, más exactas se vuelven las recomendaciones.

Finalmente, cuando se obtienen cantidades logarítmicas como representación a la probabilidad de que una película le pueda gustar al usuario o no pueda gustarle es necesario volver a tener probabilidades normales para su análisis y recomendación.

Para obtener las probabilidades normales nuevamente se procede a utilizar una fracción exponencial, en el cual se coloca en el numerador la probabilidad que se quiera obtener, por ejemplo, se colocaría en el numerador una exponencial elevada a la representación logarítmica positiva si se desea obtener la probabilidad de que al usuario le guste la película analizada. Y viceversa si se busca obtener la probabilidad de que la película no sea de su agrado. En el denominador de la fracción logarítmica se coloca la suma de las exponenciales elevadas a ambas representaciones logarítmicas, con lo cual se obtiene una probabilidad sin perdida de decimales de si al usuario le gustará o no la película evaluada.

3.5. Resultados

Como resultado final se puede observar que, conforme se interaccione en la aplicación, la misma recomienda películas que tengan similitudes con la información introducida.

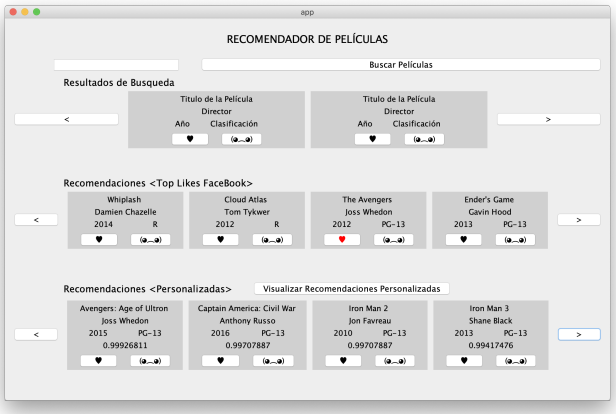

Figura 6. Voto como favorito para la película Avengers

Por ejemplo, si se introduce la película Avengers como una película favorita, las recomendaciones a partir de esa única interacción son las siguientes:

- Avengers: Age of Ultron
- Capitan America: Civil War
- Iron Man 2
- Iron Man 3

Cada una de las cuatro películas indican que tienen el 99 % y fracción de probabilidad de que le gusten al usuario. Todas las películas mencionadas cuentan con muchas factores en común, además de ser producidas por el mismo estudio, aunque no es un factor que analice el Recomendador.

A la inversa, si se vota negativo por una película como Django, al ser el único voto muestra como recomendaciones las películas que sean lo más contrarias posibles. Aún así, las películas recomendadas cuentan con unos escasos 14 % de que le gusten al usuario y menores.

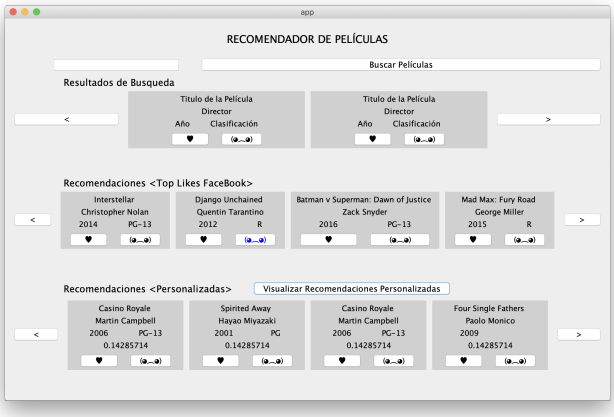

Figura 7. Voto como negativo para la película Django

3.6. Argumentación

Se tomó la decisión de utilizar Naive Bayes para la implementación del recomendador de películas, ya que es un algoritmo sin mayor complejidad y que realiza el proceso necesario con relativa facilidad lo que lo convierte en un buen candidato para llevar a cabo el desarrollo de un recomendador.

También se tomó una cantidad de datos relacionados con la película que si bien, provocan algunos segundos de espera durante la inicialización de la aplicación, son suficientes para arrojar buenos resultados en las recomendaciones y su análisis en tiempo real con la interacción al usuario son prácticamente insignificantes, por lo que brinda una buena fluidez al programa y por tanto una mejor experiencia al usuario.

4. Conclusiones

- Machine Learning es un área de mucha importancia en la actualidad, ya que gracias a la minería de datos, análisis de información, entre otros, se ha mejorado muchas industrias, como por ejemplo la industria del entretenimiento, esta industria ha mejorado sin duda ya que muchas aplicaciones ofrecen una mejor experiencia al usuario con lo cual se vuelven más populares y

exitosas, entre ellas se puede destacar a Netflix, que cada vez ofrece títulos más relacionados a nuestros gustos, lo cual provoca cierta fidelidad a la plataforma de parte de los usuarios. También se reconoce una mejora en la industria de marketing y publicidad, a través de las redes sociales, no es ningún secreto que las recomendaciones publicitarias que nos realizan las plataformas como FaceBook e Instagram llegan al usuario dados sus gustos y tendencias.

- Entre un dos algoritmos de inteligencia artificial que ofrezcan los mismos resultados, el mejor es aquel que cuente con mayor simplicidad en su desarrollo y procesos, ya que puede ser mejor utilizando los recursos del sistema, lo cuál es sumamente importante cuando la cantidad de información a analizar sea mayor.

5. Recomendaciones

- Partiendo de una buena normalización inicial de la información se puede reducir considerablemente las complejidades temporales y espaciales ya que se puede conseguir utilizar menos tiempo, espacio y recursos computacionales durante la ejecución de un algoritmo.

- Tomar en cuenta acciones de interacción del usuario en la plataforma permite comenzar con un arranque en frío de la aplicación que, dependiendo del algoritmo se pueden conseguir buenos resultados desde un inicio.

Referencias

[1] Mitchell, Tom, “Machine Learning” Ed. McGraw-Hill (1997).

[2] Fernandez, Enrique, Análisis de Clasificadores Bayesianos” Laboratorio de Sistemas Inteligentes.

[3] Felgaer, P .; Britos, P .; Sicre, J.; Servetto, A.; García-Martínez, R. y Perichinsky, G.; 2003; Optimización de Redes Bayesianas Basada en Técnicas de Aprendizaje por Instrucción.; Proceedings del VIII Congreso Argentino de Ciencias de la Computación. Pág. 1687.

[4] Ramoni, M; Sebastián P.; 1996; Learning Bayesian networks from incomplete databases. Technical report KMI-TR-43, knowledye Media Institute, The open University.

[5] Pedro Larranaga, Inaki Inza, Abdelmalik Moujahid Departamento de Ciencias de la Computacion e Inteligencia Artificial Universidad del País Vasco–Euskal Herriko Unibertsitatea