



Universidad Nacional Autónoma de México
Facultad de Ingeniería

Aprendizaje

Proyecto de Examen Extraordinario

Omar Alonso Alvarado Luna

23 de Abril de 2013

Índice

1. Introducción	3
1.1. ¿Pero qué es el Inteligencia Artificial?	3
1.2. Definiciones de Inteligencia Artificial	3
1.3. Ramas de la Inteligencia Artificial	4
1.4. Áreas de Vanguardia: Inteligencia Natural	4
1.5. El Aprendizaje Automático	4
1.5.1. Definiciones de Aprendizaje	4
1.5.2. Definiciones de Aprendizaje Automático	4
2. Descripción del Problema	5
3. Objetivo	6
4. Base teórica	6
4.1. Tipos de Aprendizaje Automático	6
4.1.1. Aprendizaje Inductivo	6
4.1.2. Aprendizaje conexionista	6
4.1.3. Aprendizaje bayesiano y adaptable	7
4.1.4. Aprendizaje genético	9
4.1.5. Aprendizaje por analogías	11
4.1.6. Aprendizaje	15
4.1.7. Etapas del razonamiento basado en casos (RBC)	16
4.1.8. Adaptación de casos	16
5. Propuesta de solución del problema	17
5.1. Conceptos básicos	17
6. Conclusiones	23
7. Bibliografía y Mesografía	23

1. Introducción

En la actualidad en nuestra vida diaria las personas están recibiendo una cantidad de información impresionante, la cual muchas veces es basura o ruido que no les interesa o no les ayuda para su quehacer diario.

Existen ya muchas aplicaciones que filtran ese tipo de información en diferentes aplicaciones que la gente utiliza para su vida o entretenimiento diario, por ejemplo, Netflix, los anuncios de Google, de Facebook, etcétera. Éstos algoritmos son en su mayor parte aplicaciones de aprendizaje automático que van entrenándose y tomando experiencia de los gustos e intereses de las personas y tratan de mostrar lo que ellos creen que es relevante para ellas.

Otro ejemplo, el Facebook y el Twitter, que son redes en las que se puede llegar a crear muchísimas conexiones con amigos o seguidores y en las cuales muchas veces están recibiendo una cantidad impresionante de basura en el muro o Timeline de la aplicación.

La cuestión aquí es poder realizar un programa de computación “Inteligente” que pueda discernir entre los diferentes entradas de twitter para saber qué es lo relevante para el usuario en específico dependiendo de su propia interacción con la Red Social, es decir, que vaya aprendiendo de tal interacción y que nos de información cada vez más útil.

Así pues se quiere hacer un un programa de computadora inteligente capaz de aprender y ser más eficiente cada vez en cuanto a su función.

1.1. ¿Pero qué es el Inteligencia Artificial?

Bien, pues éste termino parece haber sido acuñado por John McCarthy en la famosa conferencia Darmouth, donde los inicios del campo de estudio estaban siendo establecidos, definiéndose así no sólo los objetivos para la IA, sino también un amplio rango de herramientas y métodos clave para tal estudio. En lo siguiente se intentará dar una base teórica acerca de la Inteligencia Artificial y el Aprendizaje Automático, campos de estudio dentro de los cuales se encuentra la teoría que se utilizará para la realización de este proyecto.

1.2. Definiciones de Inteligencia Artificial

Hoy en día no existe una definición totalmente aceptada sobre qué es la Inteligencia Artificial, sin embargo, varios autores e interesados en el área han creado sus propias definiciones sobre IA y obviamente cda una de ellas es válida para determinado enfoque. Algunas definiciones bastante aceptables sobre el tema son las que se presentan a continuación.

- **Rich & Knight (1991)** : La inteligencia artificial es el estudio de como hacer que las computadoras hagan cosas que por el momento los humanos hacen mejor.
- **Mark Fox**: la IA es básicamente una teoría de como trabaja la mente humana.

1.3. Ramas de la Inteligencia Artificial

- **Sistemas Expertos (Sistemas basados en conocimiento).** Programas computacionales que resuelven problemas que normalmente requieren del conocimiento de un especialista o experto humano. Es un sistema capaz de tomar decisiones inteligentes interpretando grandes cantidades de datos sobre un dominio específico de problemas.[1]
- **Aprendizaje y Razonamiento Automático.** Máquinas capaces de planificar, tomar decisiones, plantear y evaluar estrategias, aprender a partir de la experiencia, autore-programables, etc.[1]
- **Robótica.** Artefactos autónomos capaces de llevar a cabo diversas tareas mecánicas de manera flexible e inteligente, cumpliendo con un objetivo y ajustándose al entorno cambiante.[1]
- **Procesamiento del Lenguaje Natural.** Sistemas capaces de reconocer, procesar y emular el lenguaje humano.
- **Visión por Computadora (Reconocimiento de patrones).** Reconoce y procesa señales, caracteres, patrones, objetos, escenas.

1.4. Áreas de Vanguardia: Inteligencia Natural

1.5. El Aprendizaje Automático

1.5.1. Definiciones de Aprendizaje

- Aprender es construir o modificar representaciones de aquello con lo que se está experimentando (McCarthy, 1968)
- El aprendizaje denota cambios en el sistema que permiten que se realice la misma tarea... más eficiente y eficazmente la próxima vez (Simon, 1983)
- El problema de definir aprendizaje se reduce a definir conocimiento. Aprendizaje sería el aumento de conocimiento (Dietterich, 1986)

1.5.2. Definiciones de Aprendizaje Automático

- Se dice que un programa aprende de la experiencia con respecto a cierta clase de tareas T y una medida del rendimiento P si la medida del rendimiento de tareas en T medidas por P aumenta con la experiencia E . (Tom Mitchell)[2]
- Campo de estudio que otorga a las computadoras la habilidad de aprender sin ser programadas de forma explícita (Arthur Samuel 1959)
- Un sistema se dice que aprende si puede modificar su comportamiento después de un conjunto de experiencias, de forma que pueda realizar la tarea con mayor precisión o más eficientemente o realizar tareas más allá de sus capacidades previas (Carbonell)

2. Descripción del Problema

Como ya se dijo en la introducción existe una aplicación llamada Twitter, que consiste en generar textos en una especie de *microblogging* que simplemente es que se pueda escribir en 140 caracteres alguna idea y ésta sea publicada en la aplicación para que los seguidores, o las personas que siguen un tema o realizan una búsqueda puedan acceder a la información de manera instantánea.

Pareciera algo sencillo y sin mayor problema si a una cuenta le siguen 100 persona, entonces al momento que esa cuenta publique una entrada de twitter, sería enviado al *timeline* de cada seguir, parece que ahí no tenemos mayor problema, sin embargo, cuando una persona sigue a 100 cuentas de twitter, o quinientas, o mil, entonces el *timeline* de éste usuario se puede volver un caos, incluso cuando se busca un tema en particular, por ejemplo **temblor** en el momento que está llevándose a cabo un terremoto en la ciudad, es impresionante la cantidad de *tuits* que pueden ser generadas, como ejemplo de la cantidad de *twits* que se puede generar en pocos momentos cito a continuación un artículo de internet:

Las redes sociales están cambiando nuestra manera de comunicarnos y relacionarnos. Hasta hace solo unos meses las audiencias median el interés general de la gente sobre ciertos temas, ahora, Twitter es una herramienta más de análisis de la sociedad. Cuál ha sido la noticia que más tuits ha generado por segundo hasta el momento.

El pasado 29 de agosto tras su actuación en los premios MTV Video Music Awards la cantante Beyoncé, mientras se acariciaba el vientre, pedía a los asistentes que le dieran la bienvenida a alguien muy especial. Esta curiosa manera de hacer público su embarazo generó una vorágine de tuits que marcaron un récord histórico en la red social de microblogging más famosa del planeta. Exactamente, en los minutos después, exactamente a las 10:35 pm, según la cuenta oficial de Twitter, se generaron 8.868 tuits por segundo.

*De este modo la noticia de la cantante superaba a otros momentos históricos en esta red social, como la catástrofe de Japón ocurrida en marzo de este mismo año que llegó a generar 5,530 tuits por segundo, la celebración del año nuevo en el país del sol naciente con 6,939 tuits al segundo y la final del mundial de fútbol femenino con 7.196 tuits por segundo.*¹

Pues aquí se puede observar una idea del potencial de información que puede tener esta aplicación, sin embargo, como ya se podrá observar, a muchas personas por ejemplo no les interesa en absoluto el embarazo de *Beyoncé*, o respuestas de usuarios que a veces se utiliza más como una sala de charla en lugar de un lugar para postear simplemente ideas.

Entonces el problema aquí es que por ejemplo un usuario puede estar siguiendo a muchas personas y su *timeline* puede contener mucho porcentaje de información totalmente irrelevante, pero puede haber entre ese monton de tuits, información que en realidad puede ser muy útil y relevante para un usuario, como fechas de maratones para un corredor, si se habla del

¹fuentes: <http://www.muyinteresante.es/tecnologia/preguntas-respuestas/icual-es-el-record-de-tuits-por-segundo-que-ha-generado-un-tema>

concierto de un artista famoso, de algún tema en particular, es por eso que el aprendizaje automático puede ser de mucha ayuda ya que éste puede aprender específicamente para un usuario lo que realmente le es importante o relevante y mostrárselo en un *timeline depurado*.

3. Objetivo

El objetivo es la realización de un programa de computadora que pueda filtrar los tuits de un timeline y presentar únicamente los que sean relevantes para el usuario.

4. Base teórica

4.1. Tipos de Aprendizaje Automático

[3]

4.1.1. Aprendizaje Inductivo

El aprendizaje inductivo puede verse como el proceso de aprender una función. Por ejemplo, en aprendizaje supervisado, al elemento de aprendizaje se le dá un valor correcto (o aproximadamente correcto) de una función a aprender para entradas particulares y cambia la representación de la función que se está infiriendo, para tratar de aparear la información dada por la retroalimentación que ofrecen los ejemplos.

Un ejemplo es un par $(x, f(x))$, donde x es la entrada (que generalmente es un vector) y $f(x)$ la salida. El proceso de inferencia inductiva pura es: dada una colección de ejemplos de f , regresar una función h tal que se aproxime a f . A la función h se le llama hipótesis. En principio existen muchas posibilidades para escoger h , cualquier preferencia se llama *bias* o *sesgo*. Todos los algoritmos de aprendizaje exhiben algún tipo de sesgo.

4.1.2. Aprendizaje conexionista

El conexionismo es un movimiento en ciencia cognitiva que trata de explicar las habilidades intelectuales humanas utilizando redes neuronales artificiales. Las redes neuronales son modelos simplificados del cerebro compuestos por gran cantidad de unidades llamadas perceptrones (que son los análogos a las neuronas) juntos con pesos que miden la fuerza de las conexiones entre las unidades. Estos pesos modelan los efectos de las sinapsis que ligan una neurona on otra. Experimentos en modelos de este tipo han demostrado la habilidad de aprender tal como el reconocimiento facial, lectura y detección de estructuras gramáticas simples.

Una descripción de las redes neuronales consiste en una gran cantidad de unidades juntas y unidas en un patron de conexiones. Las unidades en una red son usualmente segregadas dentro de 3 clases: *unidades de entrada* las cuales reciben la información para ser procesada, *unidades de salida* donde los resultados del procesamiento se encuentran, y *unidades internas* llamadas también unidades ocultas.

Cada unidad de entrada tiene un valor de activación que representa alguna característica externa a la red. Una unidad de entrada envía su valor de activación para cada una de las unidades ocultas a las cuales está conectada. Cada una de esas unidades ocultas calcula su propio valor de activación dependiendo de los valores de activación que recibe de sus unidades de entrada. Esta señal es entonces enviada hacia las unidades de salida o a alguna otra capa de las unidades ocultas. Esas unidades ocultas computan sus valores de activación en la misma forma, y las envían a través de sus vecinos. Eventualmente la señal de las unidades de entrada se propagan a través de todo el camino de la red para determinar los valores de activación de todas las unidades de salida.

El patrón de la configuración de la activación por una red es determinada por los pesos, o fuerza de las conexiones entre las unidades. Los pesos pueden ser tanto positivos como negativos. Un peso negativo representa la inhibición de la unidad que está recibiendo por la actividad de la unidad que la está enviando. El valor de activación para cada unidad de recepción es calculado de acuerdo a una simple función de activación. Las funciones de activación varían en detalle, pero todas ellas conforman el mismo plan básico.

Aprendizaje de la Red Neuronal y la Retropropagación [2] Encontrar el conjunto correcto de pesos para cumplir una tarea dada es el objetivo central de la investigación conexionista. Por suerte, los algoritmos de aprendizaje han sido ideados de tal forma que puedan calcular los pesos correctos para llevar a cabo muchas tareas. Uno de los métodos de entrenamiento más usados es llamado *retropropagación*. Para utilizar este método se necesita un conjunto de entrenamiento consistente en muchos ejemplos de entradas y sus salidas deseadas correspondiente para una tarea dada. Si por ejemplo, la tarea es distinguir una cara masculina de una femenina, el conjunto de entrenamiento puede contener fotos de caras juntas con una indicación del sexo de la persona que aparece en cada una. Una red que pueda aprender esta tarea puede tener dos tipos de unidades de salida (indicando las categorías masculino y femenino) y muchas unidades de entrada, cada una proporcionando el brillo de cada pixel en la foto. Los pesos de la red para ser entrenada son inicialmente un conjunto de valores aleatorios, y entonces los miembros del conjunto de entrenamiento son expuestos repetidamente en la red. Los valores para la entrada de un miembro son puestas en las unidades de entrada y la salida de la red es comparada con la salida deseada de este miembro. Entonces todos los pesos en la red son ajustadas cada vez en la dirección que nos pueda dar cada vez más acercado el valor de salida deseado. Por ejemplo, cuando una cara de hombre es presentada a las unidades de entrada los pesos son ajustados entonces el valor de la unidad de salida del hombre es incrementada y el valor de la unidad de salida femenino es decrementada. Después de muchas repeticiones del proceso la red puede aprender a producir la salida deseada para cada entrada en el conjunto de entrenamiento. Si el entrenamiento va bien, la red puede también aprender a generalizar el comportamiento deseado para las entradas y salidas que pueden no estar en el conjunto de entrenamiento. Por ejemplo, este puede hacer un buen trabajo para distinguir hombres de mujeres en fotos que nunca se le habían presentado antes.

4.1.3. Aprendizaje bayesiano y adaptable

El razonamiento bayesiano provee una aproximación probabilística para la inferencia. Está basado en la asunción que las cantidades de interés están gobernadas por la distribución

de la probabilidad y que las decisiones óptimas pueden ser hechas razonando acerca de estas probabilidades juntas con los datos observados. Es importante en el aprendizaje automático porque provee una aproximación cuantitativa para darle peso a la evidencia dando soporte a las hipótesis alternativas. El razonamiento bayesiano provee la base para los algoritmos de aprendizaje que manipulan directamente probabilidades, tal como una capa de trabajo para analizar la operación u otros algoritmos que no manejan explícitamente las probabilidades.

- Provee un método probabilístico para la inferencia
- Supone que las cantidades de interés son dominadas por las distribuciones de probabilidad y que las decisiones óptimas pueden ser hechas por razonamientos acerca de esas probabilidades junto con los datos observados.
- Consiste en sopesar las diferentes hipótesis y asignarles una probabilidad de acuerdo a los datos de entrenamiento que clasifica correctamente.
- Provee una perspectiva útil para entender muchos algoritmos de aprendizaje que no manejan implícitamente las probabilidades.

Características del Aprendizaje Bayesiano [2]

- Cada ejemplo de entrenamiento observado puede incrementar o disminuir la probabilidad estimada de una hipótesis.
- El conocimiento a Priori puede ser combinado con los datos observados para determinar la probabilidad final de una hipótesis
- El conocimiento a Priori es obtenido a partir de: La probabilidad para cada hipótesis candidata y la distribución de la probabilidad sobre los datos observados para cada posible hipótesis.
- Pueden adaptar hipótesis que pueden hacer predicciones probabilísticas
- Las nuevas instancias pueden ser clasificados combinando las predicciones de hipótesis múltiples, medidas por su probabilidad.
- Requiere conocimiento inicial de muchas probabilidades, cuando no se conocen deben ser estimados basándose en conocimiento de fondo
- Tiene un coste computacional elevado. En el caso general es lineal con el número de hipótesis candidatas.

Teorema de Bayes [2] En aprendizaje automático se está frecuentemente interesado en determinar las mejores hipótesis de un espacio H dado por los datos de entrenamiento D observados. Una manera de especificar lo que se quiere decir por *mejor* hipótesis es decir que queremos la *hipótesis más probable* dados los datos D más cualquier conocimiento inicial acerca de las probabilidades a priori de varias hipótesis en H . El teorema de Bayes provee una forma para calcular la probabilidad de la hipótesis basado en su probabilidad a priori, las probabilidades de observar muchos datos dada la hipótesis, y los datos observados en sí mismos.

Para definir el teorema de Bayes precisamente, primero se debe introducir una pequeña notación. Escribiremos $P(h)$ para denotar la probabilidad inicial que la hipótesis h sostiene, antes de haber observado los datos de entrenamiento. $P(h)$ es frecuentemente llamada la probabilidad *a priori* de h y puede reflejar cualquier conocimiento previo que se tenga acerca de la oportunidad de que h es una hipótesis correcta. Si no tenemos el suficiente conocimiento previo, entonces nosotros podemos simplemente asignar la misma probabilidad *a priori* para cada hipótesis candidata. Similarmente, escribiremos $P(D)$ para denotar la probabilidad *a priori* de los datos de entrenamiento D que serán observados (por ejemplo, la probabilidad de D sin haberse dado conocimiento alguno de la hipótesis que se sostiene). Posteriormente, escribiremos $P(D|h)$ para denotar la probabilidad de los datos observados D dado algún escenario en el cual la hipótesis h es sostenida. De forma más general, escribiremos $P(x|y)$ para denotar la probabilidad de x dado y . En problemas de aprendizaje automático se está interesado en la probabilidad $P(h|D)$ que h mantiene dados los datos de entrenamiento d . $P(h|D)$ es llamada la probabilidad *a posteriori* de h , porque refleja la confianza que h mantiene después de haber visto los datos de entrenamiento D . Nótese que la probabilidad *a posteriori* $P(h|D)$ refleja la influencia de los datos de entrenamiento D , en contraste con la probabilidad *a priori* $P(h)$, la cual es independiente de D .

El teorema de Bayes es la piedra angular de los métodos de aprendizaje Bayesiano porque provee una forma para calcular la probabilidad *a posteriori* $P(h|D)$, de la probabilidad *a priori* $P(h)$, junto con $P(D)$ y $P(D|h)$.

Teorema de Bayes

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

4.1.4. Aprendizaje genético

Los algoritmos genéticos proveen un acercamiento al aprendizaje que está basado en una evolución simulada. Las hipótesis son frecuentemente descritas para cadenas de bits que su interpretación depende de la aplicación, aunque las hipótesis pueden también ser descritas por expresiones simbólicas o incluso programas de computadora. La búsqueda de una hipótesis apropiada comienza con una población, o colección, o una hipótesis inicial. Miembros de la población actual dan lugar a la siguiente generación de la población por medio de operaciones tal como mutaciones aleatorias y cruces, los cuales son dibujados después del proceso de evolución biológica. En cada paso, la hipótesis en la población actual es evaluada relativo a la medida dada de aptitud, con la hipótesis más apta seleccionada probabilísticamente como semillas para producir la siguiente generación. Los algoritmos genéticos han sido aplicados exitosamente para una variedad de tareas y a otros problemas de optimización. Por ejemplo, han sido usados para aprender colecciones de reglas para el control robótico y para optimizar la topología y los parámetros de aprendizaje para redes neuronales artificiales.

Algoritmos genéticos [2] El problema abordado por AG's es la búsqueda de un espacio de hipótesis candidatas para identificar las mejores hipótesis. En AG's la "mejor hipótesis" está definida por aquella que optimiza una medida numérica predefinida para el problema en cuestión, llamado la hipótesis *idónea*. Por ejemplo, si la tarea de aprendizaje es el problema

de aproximar una función desconocida dados los ejemplos de entrenamiento de su entrada y salida, entonces idóneo puede ser definido como la exactitud de la hipótesis a través de sus datos de entrenamiento. Si la tarea es aprender una estrategia para jugar ajedrez, idóneo puede ser definido como el número de juegos ganados por el individuo cuando juega contra otros individuos en la población actual.

Aunque, implementaciones diferentes de algoritmos genéticos varían en sus detalles, ellos comparten típicamente la siguiente estructura: El algoritmo opera actualizando iterativamente un pool de hipótesis, llamadas la población. En cada iteración, todos los miembros de la población son evaluados de acuerdo a la función idónea. Una nueva población es entonces generada por una selección probabilística de los individuos más aptos de la población actual. Algunos de estos individuos seleccionados son llevados hacia la siguiente generación de la población de forma intacta. Otros son usados como base para crear nueva descendencia de individuos aplicando las operaciones genéticas tal como el cruce y la mutación.

Un algoritmo genético prototipo

AG (Aptitud, Umbral_aptitud, p, r, m)

Aptitud: Función que asigna una puntuación de evaluación dada una hipótesis

Umbral_aptitud: un umbral que especifica el criterio de terminación.

p: El número de hipótesis que serán incluidas en la población.

r: La fracción de la población que será reemplazada por el cruzamiento en cada paso.

m: La velocidad de mutación.

- Inicializar población: $P \leftarrow$ Generate p hipótesis de forma aleatoria
- Evaluar: Para cada h en P , calcular $Aptitud(h)$
- *Mientras* $[\max_h Aptitud(h)] < Umbral_aptitud$ *hacer* Crear una nueva generación, P_s :
 1. Seleccionar: Probabilísticamente seleccionar $(1 - r)p$ members of P para agregar a P_s . La probabilidad $Pr(h_i)$ de la selección de la hipótesis h_i de P es dada por

$$Pr(h_i) = \frac{Aptitud(h_i)}{\sum_{j=1}^p Aptitud(h_j)}$$

2. Cruzamiento: Seleccionar probabilísticamente $\frac{r \cdot p}{2}$ pares de hipótesis de P , de acuerdo a $Pr(h_i)$ dado arriba. Para cada par, $\langle h_1, h_2 \rangle$, produce dos descendencias por la aplicación del operador de cruzamiento. agregar toda la descendencia a P_s
 3. Mutar: Elegir m porcentaje de miembros de P_s con probabilidad uniforme. Para cada uno, invertir cada bit seleccionado aleatoriamente en su representación.
 4. Actualizar: $p \leftarrow P_s$.
 5. Evaluar: para cada h en P , calcular $Aptitud(h)$
- Regresar la hipótesis de P que tenga la aptitud más alta.

La entrada para el algoritmo anterior incluye la función de aptitud para clasificar a las hipótesis candidatas, un umbral que defini un nivel aceptable para la terminación del algoritmo de aptitud, el tamaño de la población a ser mantenida, y parámetros que determinan como serán generadas las poblaciones sucesoras: la fracción de la población a ser reemplazada en cada generación y la velocidad de mutación.

Nótese que en este algoritmo cada iteración a través del bucle principal produce una nueva generación de hipótesis basadas en la población actual. Primero, un cierto número de hipótesis de la población actual son seleccionadas para su inclusión en la siguiente generación. Estos son seleccionados *probabilísticamente*, donde la probabilidad de seleccionar la hipótesis h_i esta dada por:

$$Pr(h_i) = \frac{Aptitud(h_i)}{\sum_{j=1}^p Aptitud(h_j)}$$

Así, la probabilidad de que una hipótesis será seleccionada es proporcional a su propia aptitud e inversamente proporcional a la aptitud de las otras hipótesis que estan compitiendo en la población actual.

Una vez que esos miembros de la población actual son seleccionados para su inclusión en la siguiente generación de población, miembros adicionales son generados usando una operación de cruzamiento. El cruzamiento toma dos hipótesis padres de la población actual y crea dos hipótesis descendientes recomblando porciones de ambos padres. Las hipótesis padre son seleccionadas probabilísticamente de la población actual. Después nuevos miembros son creados por su operación de cruzamiento, la nueva generación de población ahora contiene el numero decidido de miembros. En este punto, una cierta fracción m de esos miembros son elegidos aleatoriamente, y las mutaciones aleatorias todos ellos realizados para alterara estos miembros.

Este algoritmo así crea una un haz de búsqueda paralelo y aleatorio para las hipótesis que se realizan bien de acuerdo a la función de aptitud

4.1.5. Aprendizaje por analogías

[4]

El razonamiento analógico intenta emular la capacidad humana de recordar la solución de problemas previos ante la aparición de problemas parecidos en los que se llevan a cabo razonamientos análogos para alcanzar las soluciones resepectivas.

Análisis Generalización inductiva sobre las relaciones que se cumplen en un determinado dominio (origen o fuente), suponiendo que también se cumplen en otro dominio (destino o meta).

El proceso de analogía está basado en que si dos situaciones son similares en algún aspecto entonces pueden serlo en otro. Explora la experiencia acumulada.

- **Problema Base:** El problema ya resuelto tal que su solución servirá de base para resolver el nuevo problema.
- **Conocimiento Base:** La Información disponible sobre el problema base y su dominio.
- **Problema Objetivo:** El Nuevo Problema a Resolver

Entre ambos existe una relación de causalidad.

Modelo de Razonamiento Unificado Visión unificada de los componentes básicos de un sistema analógico.

Problema Tipo. Dada como entrada una situación objetivo, da como resultado una representación aumentada de la misma en la que consten las Inferencias analógicas obtenidas de una situación base.

Fases

- **Recuperación.** Dada la situación objetivo, el sistema ha de ser capaz de recuperar un caso base potencialmente análogo y poner en correspondencia las partes correspondientes de ambas.
- **Elaboración.** Derivar atributos, relaciones o cadenas causales adicionales que pueden ser utilizadas sobre la situación objetivo
- **Mapeo.** Mapear los atributos seleccionados sobre el objetivo con posibles modificaciones
- **Justificación.** Justificar que los atributos son válidos
- **Aprendizaje.** Guardar la representación aumentada de la situación objetivo; en la creación la reglas generales motivadas por la analogía o en el refinamiento de las mismas a partir de más razonamientos sobre la misma o diferentes situaciones base

Combinación inductivo - deductivo Las “Inferencias analógicas” son un caso concreto de razonamiento analógico tienen carácter inductivo. Se fundamenta en establecer la suposición que permite relacionar las dos situaciones involucradas (s y t).

$$\frac{P(s)Q(s), P(t)}{Q(t)}$$

Parte de la idea de que si las situaciones coinciden en algunos aspectos, entonces es probable que coincidan en otros:

$$P(x) \Rightarrow Q(x)$$

Suele aplicarse sólo cuando las situaciones comparadas comparten un conjunto “suficiente” de semejanzas o cuando además la relación establecida sea relevante y sea necesario utilizar la información de la fuente para realizar la inferencia.

Problema de la no redundancia. Determinar cuáles son las suposiciones analógicas que deben hacerse. Establecer mecanismos adicionales que nos permitan acceder al conocimiento implicado en decidir en la fórmula anterior cuáles son las propiedades Q que pueden estar sujetas a inferencia analógica.

Lógica de predicados. La expresión “ P determina Q ” es representado como “ $P > Q$ ”; expresa la relación de dependencia entre las propiedades P y Q de los objetos del dominio. La nación determina el idioma.

Estas relaciones pueden utilizarse adecuadamente en un marco de programación lógica, que permite validar las inferencias realizadas en el razonamiento analógico.

Analogía transformacional Se recupera de la memoria de soluciones previas la solución del problema “más parecida al problema que se está intentando resolver. La solución recordada se toma como “Estado inicial” de un proceso de búsqueda que consistirá en aplicar una serie de operadores con vistas a alcanzar una solución del problema nuevo.

Se considera que existe un espacio (T-espacio) en el cual la solución conocida puede ser transformada, usando unos operadores (T-operadores), hasta convertirla en la solución de un nuevo problema. **Planteamiento.** Sólo se mira que la solución sea equivalente; no cómo se resuelve.

Análisis medios - fines

Espacio del Problema

- Conjunto estados posibles
- Estado inicial
- Estado final
- Conjunto de Operadores: Permiten Transformar un Estado en otro
- Función de diferencias: Computa las diferencias existentes entre dos estados dados
- Tabla de diferencias: Dada una diferencia, devuelve el operador que permite eliminarla
- Conjunto de Restricciones Globales

Resolución

1. Comparar el estado actual y el final mediante función de diferencias
2. Elegir un operador que elimine alguna de las diferencias encontradas
3. Si es posible aplicar el operador, se aplica y se obtiene un nuevo estado actual. Si no guardar el estado actual y aplicar la estrategia de resolución al problema de satisfacer dichas restricciones

4. Cuando el subproblema es resuelto, restablecer el último estado guardado y proseguir el trabajo en el problema original

No se utiliza en ningún momento Información obtenida en la resolución del primero para resolver el segundo.

Generalizar la estrategia

1. **Fase búsqueda de los precedentes.** Medida de similitud entre problemas, la función de diferencias calcula las diferencias entre estados y compara las restricciones
2. **Fase transformación de la solución.** Adaptar la secuencia de operadores en que consiste la solución del caso precedente en una que resuelva el nuevo problema y que cumpla las restricciones

Analogía Derivacional Se describe la Solución Analógica de un problema como la meta de una tarea jerárquica que almacena información detallada de la solución. El plan generado se descompone en subproblemas que indican metas intermedias; lo que permite trazar el curso de la solución.

Solución Incremental. A cada paso de la solución se resuelve un nuevo problema.

El Sistema es capaz de emplear trazas de problemas resueltos previamente. El Sistema ha de almacenar toda la información generada en cada paso, pero que dé especial importancia a aquella generada por los operadores instanciados sin analizar las razones de esa elección.

Traza

1. Dado un Problema Objetivo, el Sistema intenta Recuperar un Plan. Si resuelve el problema, Termina.
2. Elaborar una solución
3. Descomponer el problema en subproblemas
4. Mecanismo Selección de un subproblema. Si al tratar el subproblema *falla*, se almacena la secuencia y se asocia una explicación
5. La solución progresa positivamente, se selecciona operadores para el tratamiento del problema.
6. Se construye una justificación para cada Nodo

Cuando una Rama Falla, el Sistema intenta Buscar en otras Fuentes.

Aplicaciones

- Buscar nuevas soluciones a problemas ya resueltos
- Intentar optimizar las soluciones antiguas
- Método razonamiento a partir de experimentos
- La *justificación* facilita la reconstrucción de la solución y permite evaluar lo aprendido

Diferencias. No solo se transfiere entre los dos problemas implicados la traza de la solución, también se incluye cualquier decisión tomada en la solución del problema previo (Alternativas deshechadas, Razones de las decisiones, Dependencias de cualquier fuente adicional de conocimiento ...)

Razonamiento basado en casos El Razonamiento basado en casos (RBC) significa resolver problemas a partir de experiencias precedentes (Casos), adaptando soluciones antiguas para resolver problemas nuevos, o recuperando casos anteriores para iluminar aspectos de la situación actual.

Hace de su última etapa, el aprendizaje, el centro de todo el proceso analógico.

Ventajas

- RBC como técnica de adquisición de conocimiento
- RBC como mecanismo de resolución de problemas
- RBC permite trabajar en dominios de problemas de difícil estructuración y representación

Inconvenientes

- Resulta costoso en cuanto a memoria
- No ha habido trabajos extensos para la manipulación de incertidumbre

Aplicaciones Problemas legales, diagnóstico, clasificación, planificación, diseño

4.1.6. Aprendizaje

Para poder iniciar el proceso de razonamiento, es necesario disponer de un conjunto de casos precedentes (Librería de casos).

Los casos pertenecen a un determinado dominio que es necesario conocer para poder interpretar la información contenida en ellos.

Cada nuevo caso resuelto por el sistema presenta una nueva ocasión para aprender.

Tipos

1. Aprender nuevos casos que ayude a evitar problemas
2. Aprender características que predigan problemas
3. Aprender reparaciones aplicables en diversas situaciones

4.1.7. Etapas del razonamiento basado en casos (RBC)**Recuperación de casos**

Determina cuando dos casos (C_m en Memoria y C el Caso Actual) son semejantes.

Diferentes tipos de similitud:

- Estructural (Forma)
- Semántica (Significado)
- Organizativa
- Pragmática

Para determinar si un caso es similar a otro, se define una función de comparación (Matching) donde se confrontan las características de un caso con las del otro. Informa si la similitud es exacta (idénticos) o parcial (determinada por un grado de semejanza). Se pueden utilizar un conjunto de heurísticas que ayuden a determinar que características son más relevantes.

Si un conjunto de casos en memoria obtienen un resultado parcial se elige un solo caso de entre todos ellos.

4.1.8. Adaptación de casos

Acomodar la solución del caso C_m recuperado de la memoria para obtener la solución del caso C . Es necesario tener en cuenta las diferencias.

Tipos

- Adaptar la solución.
 - Reinstalación o substitución de una característica por otra
 - Búsqueda local de características en una red de discriminación
 - Uso de “Críticos” que definen un conjunto especial de reglas de adaptación para manipular características que interaccionan
 - Eliminación de una característica espuria (No útil)

5. Propuesta de solución del problema

Básicamente tenemos que plantear el problema en términos de las variables que vamos a utilizar para el aprendizaje del programa, para el caso del éste, vamos a definir los conceptos que nos serán útiles para el programa:

5.1. Conceptos básicos

- **twitter_account (Cuenta de twitter):** Es la cuenta del usuario que tendrá el programa *inteligente* y se presenta en twitter con una arroba seguida del pseudónimo del usuario (Ejemplo: @superusuario @pedroperez)
- **Tweet (tuit):** Son los textos de 140 caracteres que el usuario de twitter publica en su cuenta de twitter, puede contener *mentios*, *hashtags* o *url's*.
- **Retweet (retuit):** Es volver a publicar un tuit de otro usuario para que sea visible para los seguidores de la cuenta de twitter que realiza esta acción.
- **Mention (mención):** Se trata de citar a un usuario de twitter con el arroba respectivo para que públicamente se le retransmita éste mensaje.
- **Hashtag (tema):** Es una etiqueta que comienza con el caracter # y se escribe sin espacios, creando una especie de temática sobre la cual se desea hablar y hacer público de forma que sea visible para los usuarios que estén buscando sobre el tema no importando que sea un seguidor o no, por ejemplo, si quisiera hablar sobre el tema de *#robots* tendría que mencionar este tema en mi publicación y así se puede crear una lista de mensajes que contengan ese tema.
- **Followers (seguidores):** Son todas aquellas cuentas de twitter que siguen a un usuario y de tal forma que las publicaciones de éste último les sean visibles al *follower*, regularmente se sigue a personas que se conocen ambas, que tienen mensajes interesantes o que simplemente son personas públicas y el usuario de twitter está interesado en lo que pueda publicar.
- **Following (siguiendo):** Son los usuarios de twitter a los cuales la cuenta de twitter sigue, es decir, que todas las publicaciones que envíen los usuarios que se está siguiendo, serán vistos en el *timeline*.
- **Timeline (línea de tiempo):** Es la página que contiene los tuits de los usuarios a los que está siguiendo la cuenta de twitter y están ordenados de forma cronológica.
- **DM (Mensajes directos):** Son publicaciones que sólo están visibles para el usuario al que se le está enviando en particular, es decir son privados por defecto.
- **List (Lista):** Es una agrupación de usuarios de twitter que siguen un criterio lógico. El objetivo de estas listas es organizar cuentas de usuario según temáticas específicas, por ejemplo: *corredores*, *compañeros de trabajo*, *genios*, *campus party*, *etc.*

Ahora bien, para que se pueda crear una lista de tuits relevantes, se requiere calificar cada tuit ya sea del *timeline*, o de un *hashtag* o incluso de todos los tuits en general, por lo que se necesitarán variables del tuit que nos permitan valorar su relevancia y decidir si se muestran o no en la lista de relevantes para el usuario.

Debido al tiempo de elaboración, se intentará plantear claramente el problema y la solución en base a un número aceptable de variables, y con valores tratables para el desarrollo del problema. Entonces buscaremos clasificar si un tuit es relevante o no, de manera probabilística y utilizando el *Aprendizaje Bayesiano*, con un conjunto de datos de entrenamiento, y haciendo que las variables sean discretas, a continuación se presentan las variables a utilizar.

1. **es_mencion:** Si es una mención entonces automáticamente será relevante, sería parte del conocimiento previo del programa.
2. **tiene_hashtags_relevantes:** este es un sí o no, dependiendo de una función que no veremos en este tema, pero que buscaría en una base de datos si tiene hashtags que el usuario sigue, o le han sido relevantes con anterioridad.
3. **tiene_palabras_relevantes:** aquí también se tendría un valor booleano de acuerdo a una función que nos diría si contiene palabras que el usuario ha marcado como relevantes con anterioridad:
4. **seguimiento:** Aquí tendremos cuatro posibles valores: 1) No hay seguimiento, 2) es seguidor, 3) se sigue, 4) se sigue y es seguidor

La variable que nos dará el resultado estará basado en cuatro acciones que puede llevar a cabo la cuenta de twitter que son:

- **retweet:** Si el usuario da un retweet a un mensaje, entonces la publicación tendría que pasar por una función que incremente la relevancia de las palabras o hashtags contenidas, y se haga una retroalimentación a la base de datos de información relevante para el usuario.
- **response:** Si el usuario responde a un mensaje, también se convierte en relevante y de la misma forma pasará por una función que incremente los pesos de las palabras relevantes en la base de datos, incluso éste incremento puede ser diferente para cada caso, dependiendo si es más relevante un *retweet*, un *response*, o un *favorite*.
- **favorite:** Un usuario puede marcar como favorito un tuit, y este se convierte en relevante.

Se utiliza este tipo de aprendizaje ya que primero se intentaba abordar el problema con el *Gradiente de Descenso* pero debido a que son muchas variables, y para acercarse a un mínimo tendrían que realizarse demasiados cálculos, y podría tornarse muy lento. Posteriormente se vieron las ventajas de utilizar el *aprendizaje bayesiano* y se adaptó el problema para que se pudiera abordar el problema de ésta forma.

Algunas características del *razonamiento bayesiano* son las siguientes:

- Nos da un enfoque probabilístico de la inferencia
- Está basado en asumir las incógnitas de interés siguiendo distribuciones probabilísticas.
- Se puede conseguir una solución óptima por medio de estas distribuciones y datos observados.
- Nos da la posibilidad de realizar una ponderación de la posibilidad de ocurrencia de una hipótesis de manera cuantitativa.[5]

Importancia del razonamiento bayesiano [5]

- Los algoritmos de aprendizaje bayesiano pueden calcular probabilidades explícitas para cada hipótesis.
- También nos proporcionan un marco para estudiar otros algoritmos de aprendizaje.

El aprendizaje se puede ver como el proceso de encontrar la hipótesis más probable, dado un conjunto de ejemplos de entrenamiento D y un conocimiento *a priori* sobre la probabilidad de cada hipótesis.

Característica

- Cada ejemplo de entrenamiento afecta a la probabilidad de las hipótesis. Esto es más efectivo que descartar directamente las hipótesis incompatibles.
- Se puede incluir conocimiento *a priori*: probabilidad de cada hipótesis; y la distribución de probabilidades de los ejemplos.
- Es sencillo asociar un porcentaje de confianza a las predicciones, y combinar las predicciones en base a su confianza.
- una nueva instancia es clasificada como función de la predicción de múltiples hipótesis ponderadas por sus probabilidades.
- Incluso en algunos casos en los que el uso de estos métodos se ha mostrado imposible, pueden darnos una aproximación de la solución óptima.

Dificultades:

- Necesidad de un conocimiento *a priori*. Si no se tiene este conocimiento estas probabilidades han de ser estimadas.
- Coste computacional alto. En el caso general es lineal con el número de hipótesis candidatas.

Teorema de Bayes Pondremos nuevamente la fórmula del teorema para recordar los puntos importantes.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

donde:

- $P(h)$ es la probabilidad *a priori* de la hipótesis h .
- $P(D)$ es la probabilidad de observar el conjunto de entrenamiento D .
- $P(D|h)$ es la probabilidad de observar el conjunto de entrenamiento D en un universo donde se verifique la hipótesis h .
- $P(h|D)$ es la probabilidad *a posteriori* de h , cuando se ha observado el conjunto de entrenamiento D .
-

Selección de hipótesis En algunos casos la multiplicación por $P(h)$ no tiene sentido ya que las hipótesis son equiprobables.

$$h_{ML} \equiv \underset{h \in H}{\operatorname{argmax}} P(D|h)$$

a este resultado se le denomina máxima verosimilitud.

Ejemplo ¿Un paciente está enfermo? Un test de laboratorio ha dado positivo. Cuando el paciente está enfermo el test lo detecta en un 98 % de los casos. Si el paciente no tiene cáncer, el test da un 3 % de falsos positivos. Sólo el 0.8 % de las personas están enfermas.

- $P(enfermo) = 0,008$
- $P(noenfermo) = 0,992$
- $P(+|enfermo) = 0,98$
- $P(-|enfermo) = 0,02$
- $P(+|noenfermo) = 0,03$
- $P(-|noenfermo) = 0,97$

Aplicando map para no enfermo en caso de que de test positivo

$$P(+|enfermo)P(enfermo) = 0,09 * 0,008 = 0,0078$$

$$P(+|noenfermo)P(noenfermo) = 0,03 * 0,992 = 0,0298$$

normalizando:

$$P(enfermo|+) = \frac{0,0078}{0,0078 + 0,0298} = 0,21$$

Aprendizaje de hipótesis por fuerza bruta Dado un conjunto de ejemplos D y un espacio de hipótesis H .

1. Para cada hipótesis h perteneciente a H se computa:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Se devuelve la hipótesis con la máxima probabilidad *a posteriori*

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h|d)$$

Clasificador bayesiano naive Este es el método que utilizará el algoritmo de aprendizaje de twitter por las características que vamos a ver a continuación.

- Uno de los mejores métodos de aprendizaje en la práctica
- En algunos dominios comparable a redes de neuronas y árboles de decisión.
- Se puede aplicar cuando:
 - Se dispone de conjuntos de entrenamiento de tamaño medio o grande, especialmente para conjuntos grandes de post como los de Twitter.
 - Los atributos que describen a los ejemplos son independientes entre sí con respecto al concepto que se pretende aprender.
- Aplicado con éxito en: Diagnósticos, Clasificación de documentos.
- Cada ejemplo x se describe con la conjunción de los valores de sus atributos $\langle a_1, a_2, \dots, a_n \rangle$
- La función objetivo $f(x)$ puede tomar cualquier valor de un conjunto finito V
- La clasificación viene dada por el valor de máxima probabilidad *a posteriori* V_{MAP}

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2, \dots, a_n)$$

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n) P(v_j)}$$

- Los términos se han de estimar basándose en los ejemplos de entrenamiento.
 - $P(v_j)$ contando la frecuencia con la que ocurre cada valor v_j
 - Hay demasiados términos de la forma $P(a_1, a_2, \dots, a_n | v_j)$. Harían falta muchísimos ejemplos de entrenamiento para obtener una buena estimación.
 - La suposición del clasificador naive es que los atributos son independientes entre sí con respecto al concepto objetivo y , por lo tanto

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j)$$

- la aproximación del clasificador bayesiano naive es:

$$v_{nb} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

- Las probabilidades $P(a_i | v_j)$ resultan mucho más fáciles de estimar que las $P(a_1, a_2, \dots, a_n)$

#tuit	es mención	tiene hashtags relevantes	tiene palabras relevantes	seguimiento	es relevante
1	yes	yes	no	1	yes
2	no	yes	no	2	no
3	yes	yes	yes	4	yes
4	no	yes	yes	1	no
5	yes	no	yes	2	no
6	no	no	no	3	yes
7	yes	no	yes	4	yes

Cuadro 1: Muestra los datos de entrenamiento y el resultado hipotético

Algoritmo

Aprendizaje_Bayesiano_Naive(Ejemplos)

Para cada posible valor del resultado v_j Obtener estimación $P'(v_j)$ de la probabilidad $P(v_j)$ Para cada valor a_i de cada atributo a Obtener estimación $P'(a_i|v_j)$ de la probabilidad $P(a_i|v_j)$

Clasificar_instancia(x)

devolver $v_{nb} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$ **Estimación de probabilidades**

$$P'(es_relevante = yes) = 4/7 = 0,57 \quad P'(es_relevante = no) = 3/7 = 0,43$$

- $P'(es_mencion = yes|es_relevante = yes) = 3/4 = 0,75$
- $P'(es_mencion = yes|es_relevante = no) = 1/3 = 0,33$
- $P'(tiene_hashtags_relevantes = yes|es_relevante = yes) = 2/4 = 0,5$
- $P'(tiene_hashtags_relevantes = yes|es_relevante = no) = 2/3 = 0,66$
- $P'(tiene_palabras_relevantes = yes|es_relevante = yes) = 2/4 = 0,5$
- $P'(tiene_palabras_relevantes = yes|es_relevante = no) = 2/3 = 0,66$
- $P'(seguimiento = 1|es_relevante = yes) = 1/4 = 0,25$
- $P'(seguimiento = 1|es_relevante = no) = 1/3 = 0,33$
- $P'(seguimiento = 2|es_relevante = yes) = 0/4 = 0,00$
- $P'(seguimiento = 2|es_relevante = no) = 2/3 = 0,66$
- $P'(seguimiento = 3|es_relevante = yes) = 1/4 = 0,25$

- $P'(\text{seguimiento} = 3 | \text{es_relevante} = \text{no}) = 0/3 = 0,00$
- $P'(\text{seguimiento} = 4 | \text{es_relevante} = \text{yes}) = 2/4 = 0,50$
- $P'(\text{seguimiento} = 4 | \text{es_relevante} = \text{no}) = 0/3 = 0,00$
- ...

Ejemplo a clasificar:

(es_mencion = yes, tiene_hashtags_relevantes = no, tiene_palabras_relevantes = yes, seguimiento = 2)

$$v_{nb} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

$$\underset{v_j \in V}{\operatorname{argmax}} P(v_j) P'(\text{es_mencion} = \text{yes} | v_j) P'(\text{tiene_hashtags_relevantes} = \text{no} | v_j) P'(\text{tiene_palabras_relevantes} = \text{yes} | v_j)$$

6. Conclusiones

Después de haber estudiado y visto algunos ejemplos de la metodología para desarrollar algoritmos para el aprendizaje automático se ha podido observar que gran parte de la buena aplicación de estas técnicas está en la forma en que se plantea el problema, ya que por ejemplo, al querer aplicar el método de gradiente de descenso para encontrar una función que se aproximara a los ejemplos se tornaba extremadamente complicado tratar con cientos de datos como son las entradas de tuit que como vimos pueden llegar a ser cientos en un segundo y trabajarlas para después calcular el mínimo error sería muy difícil siquiera pensarlo.

Por otro lado, se presentaba la dificultad de tener varios posibles resultados que podrían retroalimentar el conocimiento previo del programa, porque podríamos tener como respuesta una marca de favorito, una respuesta, una mención, los cuales indican que automáticamente son relevantes. Lo que se propuso fue entonces tomar en cuenta cualquiera de estas respuestas para clasificar el tuit como relevante.

7. Bibliografía y Mesografía

Referencias

- [1] Alberto Pacheco, *Inteligencia Artificial y Sistemas Inteligentes*, <http://www.depi.itch.edu.mx/apacheco/ai/index.htm>
- [2] Tom M. Mitchell, *Machine Learning*. Mc Graw Hill, USA, 1997.
- [3] Eduardo Morales y Jesús González, *Aprendizaje Computacional*, <http://ccc.inaoep.mx/jagonzalez/ML/principal/principal.html>
- [4] Alberto Pesquera Martín, *Aprendizaje en Inteligencia Artificial*, <http://www.sindominio.net/apm/articulos/ia/>

- [5] Oscar Prieto y Raúl Casillas, *Aprendizaje Bayesiano*,
<http://www.infor.uva.es/isaac/doctorado/AprendizajeBayesiano.pdf>