ISSN 0121-1129

Rough Sets, una técnica de inteligencia artificial basada en el Modelo de Datos Relacional

Rough Sets, an Artificial Intelligent Technique Based upon the Relational Data Model

Beitmantt Geovanni Cárdenas Quintero*

Resumen

Muestra la trayectoria que tiene la información desde su presentación más primitiva, los procesos a los que se ve abocada para dar cumplimiento a los objetivos de los sistemas de información de los cuales hace parte y la transformación a la que es sometida para su almacenamiento, procesamiento, extracción y entendimiento. Dentro de esto último, destaca una de las técnicas de Inteligencia Artificial más innovadoras en la manipulación y depuración de datos para el ingreso a sistemas de información, Rough Sets o Conjuntos Aproximados, especificando la similitud y aplicabilidad en sus aspectos básicos con los aspectos que tienen que ver en el desarrollo de dichos sistemas, pero basados en el Modelo de Datos Relacional. Además de esto se encontrarán definiciones y componentes propios de este modelo matemático.

Palabras clave: Inteligencia artificial, Modelo de Datos Relacional, Conjuntos Aproximados, Rough Sets.

Abstract

It shows the path that has to go through the information from its most primitive presentation, the processes that has to undergo in order to fulfil the information systems' objectives of which is part, and the conversion that it goes through for storage, processing, extraction and understanding. Highlighting the latter step, Rough Sets or approximate sets, is one of the most innovative artificial intelligence' techniques, to handle and purify the data entrance into the information systems, which woks by specifying the similarity and applicability of their basic aspects involved in the development of such systems, based on the Relational Model of Data. In addition were added some definitions and components of this mathematics model.

Key words: Artificial Intelligence, Relational Data Model, Approximate Sets, Rough Sets.

* Universidad Distrital "Francisco José de Caldas" Bogotá.
Correo e.: beitmantt@yahoo.com

1. Introducción

La Ingeniería de Sistemas es la rama de las ingenierías que proyecta un profesional capaz de desarrollar y administrar soluciones y servicios informáticos de clase mundial, que faciliten la toma de decisiones estratégicas, tácticas y operativas para que las organizaciones puedan obtener una ventaja competitiva sostenible. En sus áreas de aplicación se tendrá que analizar, diseñar, construir, implementar y soportar, administrar, simular y modelar matemáticamente, investigar y emprender las posibles soluciones a necesidades que se presentan en la sociedad, en la respectiva área de trabajo o esfera de actuación; ya que el ingeniero de sistemas se enfrenta cada día a una gran cantidad de información, donde el usuario que la genera es algunas veces consciente de que lo hace y otras veces inconsciente de ello porque lo desconoce.

2. Trayectoria de los datos

En sí, nos damos cuenta de que generamos información cuando registramos nuestra entrada en el trabajo, cuando entramos en un servidor para ver nuestro correo, cuando pagamos con una tarjeta de crédito o cuando reservamos un tiquete de avión. Otras veces no nos damos cuenta de que generamos información, como cuando conducimos por una vía donde están contabilizando el número de automóviles que pasan por minuto, cuando se sigue nuestra navegación por Internet o cuando nos sacan una fotografía del rostro al haber pasado cerca de una oficina gubernamental.

¿Y qué con todo esto? Pues que con lo anterior vienen problemas que los ingenieros de sistemas deben atender, ante tanto flujo de información y en grandes cantidades, como se puede notar. Para el caso de la *representación de información,* la Ingeniería de Sistemas se ha provisto de técnicas de modelamiento de datos para que la percepción del mundo pueda ser descrita como una sucesión de fenómenos. Desde el comienzo de los tiempos el hombre ha tratado de descubrirlos, ya sea que los entienda completamente o no. Es aparente que una interpretación del mundo es necesaria, la que debe ser suficientemente abstracta para que no sea afectada por la dinámica del mundo (los pequeños cambios), y debe ser suficientemente robusta para poder representar cómo los datos y el mundo se relacionan. Una herramienta como esta es llamada modelo de datos, el cual permite representar en forma más o menos razonable alguna realidad. El modelo de datos permite realizar abstracciones del mundo, permitiendo centrarse en los aspectos macros, sin preocuparse de las particularidades; así nuestra preocupación se centra en generar un esquema de representación, y no en los valores de los datos; algunas de estas alternativas son, por ejemplo, el Modelo Entidad-Relación (E/R) y Modelo Relacional [1, 2, 3] y las extensiones del modelo E/R, como son los modelos orientados a objetos [4], UML [5], The Unified Software Development Process [6], entre otras.

Por otra parte, el *almacenamiento y administración* de dicha información ha tenido soporte en los Sistemas Gestores de Bases de Datos (SGBD), donde el mercado de manejadores de bases de datos es bastante grande y ofrece demasiadas alternativas a la hora de elegir un software en qué confiar. En el momento de tomar una decisión con respecto a ¿por cuál herramienta inclinarnos?, ¿cuál es la óptima?, ¿cuál me ofrece mayores garantías en mi desarrollo específico?, ¿qué detalles de implementación debemos tener en cuenta para elegir nuestro sistema de gestión de bases de datos? se convierte en una gran preocupación y responsabilidad conocer las características, ventajas y desventajas, pues no desconocemos que las herramientas constituyen un aspecto fundamental a la hora de desarrollar un proyecto o una implementación; las características de los proyectos, de las compañías o las necesidades hacen prioritario que estas herramientas se ajusten a esos requerimientos específicos; por estas razones se debe profundizar e investigar las diferentes alternativas que se tienen al alcance, evitando inconvenientes posteriores como son pérdida de tiempo, pérdida de dinero o, aún más grave, comprometer la credibilidad profesional al avalar un concepto técnico sin el suficiente soporte y conocimiento. Algunos de los sistemas gestores de bases de datos que encontramos son ORACLE [www.oracle.com] y SQL Server de Microsoft

[www.microsoft.com], My SQL SERVER [www.mysql.com] y Postgresql [www.mysql.org], entre otros.

3. Implicación de la inteligencia artificial

Finalmente nos encontramos con el gran reto de que, tras de haber construido herramientas como las mencionadas anteriormente, es necesario implementar técnicas para la extracción de información, y, aún más, que la información extraída lleve consigo la posibilidad de generar conocimiento. Lo que se pretende con esta solución es descubrir conocimiento *oculto* a partir de grandes volúmenes de datos. Desde la década pasada, debido a los grandes avances computacionales, se han ido incorporando estas técnicas a las organizaciones para constituirse en un apoyo esencial al momento de tomar decisiones; hablamos de "data mining", el cual surge como una tecnología que intenta ayudar a comprender el contenido de una base de datos. De forma general, los datos son la materia prima bruta. En el momento que el usuario les encuentra algún significado estructural o funcional pasan a convertirse en información. Cuando los ingenieros de sistemas elaboran o encuentran un modelo, y la interpretación resultante de la confrontación de dicho modelo con la información represente un valor agregado, entonces se puede decir que se ha generado conocimiento. Fase importante en las actividades anteriores es el pre-procesamiento de datos, tomándose como el conjunto de procedimientos y técnicas que buscan extraer patrones dentro de un conjunto de datos [7]. Dichas técnicas tienen diversas fundamentaciones, en este caso particular nos vamos a centrar en una de las técnicas de inteligencia artificial, la cual ofrece alternativas de reducción de costos computacionales en nuestros aplicativos.

Uno de los problemas presentes cuando se realiza la solución de problemas con técnicas de inteligencia artificial, usando datos y no conocimiento explícito, es lograr trabajar con la menor cantidad posible de información sin perder calidad en la solución que se encuentre. Aquí es donde nace la propuesta del investigador polaco Zdzislaw Pawlak [8], la Teoría de Rough Sets (en adelante TRS), que se ha venido desarrollando desde la década de los ochenta; a través de todo este tiempo, como toda teoría, se ha ido enriqueciendo con nuevos aportes, derivados de una mayor investigación sobre sus alcances y bondades, tanto para aplicaciones teóricas como prácticas. Una de las aplicaciones representativas de esta teoría es el *Preprocesamiento* de los datos; se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos, siendo esta etapa de gran importancia, ya que en esta se consume generalmente alrededor del setenta por ciento del tiempo total de un proyecto de "data mining".

4. Teoría Rough Sets o Conjuntos Aproximados

Hoy la TRS ha evolucionado hasta convertirse en una metodología para enfrentar una amplia variedad de problemáticas, entre ellas la incertidumbre motivada por información incompleta o imprecisa. La TRS parece ser un modelo matemático natural para la vaguedad y la incertidumbre. La vaguedad es una propiedad de los conjuntos (conceptos) y puede ser atribuida a los límites del conjunto, mientras que la incertidumbre es una propiedad de los elementos de un conjunto y tiene que ver con pertenencia o no de estos.

Un ejemplo es cuando se descubre conocimiento, en el cual esta necesidad se manifiesta en dos sentidos: por una parte es útil tener que considerar la menor cantidad posible de datos sobre los que trabajar para realizar el aprendizaje, y segundo, es también necesario generar conocimiento explícito, por ejemplo, reglas, con la menor cantidad de atributos posibles, siempre que se mantenga la misma eficiencia y eficacia en el proceso. En ese sentido, la TRS se basa en el concepto de "indiscernibilidad". Considerando que indiscernir significa no conseguir distinguir una cosa de otra por medio de los sentidos o de la inteligencia humana; lo que busca la TRS es encontrar todos aquellos objetos (acciones, alternativas, candidatos, pacientes, etc.) que producen un mismo tipo de información, es decir, aquellos objetos que son "indiscernibles" [9]. A partir de este concepto es que entonces se generan las bases de matemáticas de esta teoría. Así, el presente trabajo

busca mostrar los alcances de esta referenciando trabajos actuales basados en TRS, y mediante ejemplos prácticos definir los componentes que hacen parte de su esencia.

4.1 *Componentes de TRS*

4.1.1 Objetos: son *entidades* que tratamos en nuestra vida; es muy fácil identificar en los objetos del mundo real sus características principales, es decir, las propiedades que los hacen diferentes entre sí, o las acciones que puede realizar [1]; es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo. Actualmente se usan para estructurar objetos materiales, servicios, conceptos, en general, cualquier abstracción mental. Tenemos algunos ejemplos: son todas aquellas acciones, alternativas, candidatos, pacientes, empresas, etc., una estructura algebraica (ej. un grupo), un ámbito de trabajo (ej. colores), un objeto material (ej. un satélite), un servicio inmaterial (ej. www), algo muy "real": una cuenta corriente.

Aprender cómo clasificar objetos en una de las categorías o clases previamente establecidas es una característica de la inteligencia de máximo interés para investigadores tanto de psicología como de informática, dado que la habilidad de realizar una clasificación y de aprender a clasificar otorga el poder de tomar decisiones.

Aplicación en Rough Sets. En TRS los objetos son dispuestos dentro de la estructura de una matriz que más adelante se explicará en detalle, por ahora su presentación se visualiza así:

Tabla 1. Disposición de los objetos en la matriz en TRS

Objetos					
Objeto-1					
Objeto-2					
Objeto-3					
…					
Objeto-n					

Tabla 2. Ejemplo de disposición de los objetos en la matriz en TRS

Docentes				
Docente-1				
Docente-2				
Docente-3				
…				
Docente-n				

4.1.2 ***Atributos***. Es una unidad básica e indivisible de información acerca de una ***entidad*** [10]. Por ejemplo, la entidad proveedor tendrá los atributos nombre, domicilio, e-mail, NIT. Algunos atributos de un conjunto de ***entidades*** son especiales. De partida es necesario definir cuáles de ellos son opcionales y cuáles son obligatorios. Como el nombre lo indica, un atributo obligatorio es aquel que siempre debe estar definido para toda entidad (ej. el NIT del proveedor). Un atributo opcional, en cambio, puede quedar sin definir para algunas de las entidades del conjunto de entidades (ej. el e-mail de un proveedor). En general, es deseable que la mayor cantidad de atributos posible se definan como obligatorios, puesto que permite simplificar mucho algunas operaciones, al tiempo que asegura una mejor integridad de los datos.

En la bibliografía explorada se encuentran tres algoritmos de aprendizaje clasificadores para comparar los efectos de la selección de atributos; uno probabilística Naïve Bayes [11], otro basado en las técnicas de vecinos más cercanos [12] y un tercero basado en árboles de decisión [12]. Estos se caracterizan por ser representativos de diferentes tipos de clasificadores, y se usan con frecuencia en los estudios comparativos y en bastantes aplicaciones de minería de datos [13].

Aplicación en Rough Sets. En TRS se contempla el conjunto de atributos notado como A, y este a la vez se divide en el conjunto de atributos Condicionales C y el conjunto de atributos de Decisión D, lo que nos lleva a:

$A = C \cup D$, donde la unión de C y D constituyen A *y* $C \cap D = \varnothing$, la intersección de C y D es vacía.

Definiendo aún más en detalle, los ***atributos condicionales*** son aquellos que permiten describir las características de un objeto de un universo dado, por ejemplo, el color, la textura, el olor de un elemento. Mientras que un ***atributo de decisión*** permite agrupar los objetos de un universo según algún criterio dado, por ejemplo, una clase en proceso de clasificación, un valor de decisión si el objeto describe un estado o situación, el estrato de una zona residencial, el tipo de contratación de un grupo de empleados.

Tabla 3. Disposición de los atributos en la matriz en TRS

Objetos	**Atributo 1**	**Atributo 2**	**...**	**Atributo n**
Objeto-1				
Objeto-2				
Objeto-3				
...				
Objeto-n				

Tabla 4. Ejemplo de disposición de los atributos en la matriz en TRS

Docentes	**Nivel de Estudios**	**N.° de cursos**	**Nivel de Evaluación**	**Nivel de Investigación**	**Buen Docente**
Docente-1					
Docente-2					
Docente-3					
...					
Docente-n					

- Objetos (docentes). Es un conjunto de objetos llamados el universo.
  U = {1, 2, 3, 4, 5, 6}

- Atributos de Condición
  C = {Nivel de estudios, N.° de cursos, Nivel de evaluación, Nivel de investigación}

- Atributos de Decisión
  D = {Buen docente}

- Atributos
  Q = {Nivel de estudios, N.° de cursos, Nivel de evaluación, Nivel de investigación, Buen docente}

4.1.3 *Dominio*. Sea f una función

$$f: X \rightarrow Y$$

El conjunto X es el dominio de definición de f. Llamamos dominio de definición de una función al conjunto de existencia de dicha función, es decir, los valores para los cuales la función está definida. El conjunto Y es el recorrido o imagen de f, que es el conjunto de valores que se obtienen a partir del dominio de definición [2]. En general, los dominios serán más bien amplios, aunque cuando se lleva a cabo la implementación es preferible restringir los dominios lo más posible, de manera que el sistema de información automáticamente haga algunas verificaciones sobre los datos que se almacenan, para asegurar la integridad de los datos.

Por ejemplo, la función $f(x) = \sqrt{x}$ tiene por ***dominio*** al conjunto de los números reales mayores o iguales que cero, ya que la raíz de números negativos no se puede calcular.

Aplicación en Rough Sets. Según lo anterior, se puede decir que es el conjunto de valores que puede tomar cada atributo. En palabras más prácticas, para el ejemplo de los docentes del concepto anterior, se puede establecer el dominio de estos de acuerdo con el atributo que determina su tipo de contratación, el dominio de los Docentes de Planta, el dominio de los Docentes Ocasionales, el dominio de los Docentes Catedráticos, son tres conjuntos diferentes, ya que los elementos que los conforman son distintos, y por lo tanto las acciones u operaciones ejercidas sobre estos pueden variar por el alcance de cada conjunto. Esto lleva en la TRS a determinar un • , como un conjunto finito de objetos, que para el momento del procesamiento de datos se toma como el universo de la función.

La notación que representa dichos posibles valores es:

Ejemplo

Atributo	Dominio del Atributo
Nivel de estudios	Pregrado, Especialista, Magíster, Doctor, PHD
N.° de cursos	1 a 7
Nivel de evaluación	Baja, Normal, Alta, Muy Alta
Nivel de investigación	Baja, Normal, Alta, Muy Alta
Buen docente	Si, No

4.1.4 *Valores*. En los modelos de datos relacionales cada atributo simple tiene un conjunto de valores o dominio asociado, que especifica el conjunto de valores que puede asignarse a cada entidad individual. Por ejemplo, si las edades de los empleados pueden variar entre 16 y 70, entonces el dominio de Edad es:

$$\{x \in N / 16 <= x <= 70\}$$

El valor del atributo A para la entidad e es A(e). Un valor nulo se representa por el conjunto vacío. Para un atributo compuesto A, el dominio V es el producto cartesiano de P(V1), ..., P(Vn) donde V1, ..., Vn son los dominios de los atributos simples que forman A:

$$V = P(V1) \times P(V2) \times ... \times P(Vn)$$

Notemos qué atributos compuestos y multivalorados pueden ser anidados de cualquier manera. Podemos representar anidamiento agrupando componentes de un atributo compuesto entre paréntesis, separando componentes con comas, y mostrando atributos multivalorados entre llaves.

Ejemplo: Si una persona puede tener más de una dirección, y en cada una de ellas hay múltiples teléfonos, podemos especificar un atributo DirTel para una PERSONA así:

{DirTel ({Teléfono (Código Área, NúmTel) }, Dirección (DirCalle (Calle, Número, NúmDepto), Comuna, Ciudad, Región)) }

La persona Beitmantt Geovanni Cárdenas Quintero puede tener una instancia de este atributo así: {DirTel ({Teléfono (7, 442-285) }, Dirección (DirCalle (Avenida Norte, 66, 302-B20), Parques del Nogal, Tunja, Boyacá)).

Aplicación en Rough Sets. Como se vio en el apartado anterior, los valores de los atributos no están dados por el dominio de ellos, sino que en este caso se hace la aclaración de manera específica, ya que su notación así lo requiere.

Ejemplo

$V_{\text{Nivel de estudios}} = V_n$ = {Pregrado, Especialista, Magíster, Doctor, PHD}... Dominio del atributo

$V_{\text{Numero de Cursos}} = V_c = \{0,1,2,3,4,5,6,7,8,9\}$

$V_{\text{Nivel de Evaluación}} = V_e$ = {Baja, Normal, Alta, Muy Alta}

$V_{\text{Nivel de Investigación}} = V_i$ = {Baja, Normal, Alta, Muy Alta}

$V_{\text{Buen Docente}} = V_d$ = {Si, No}

4.1.5 ***Tablas de Información.*** Lo que tradicionalmente han sido siempre los archivos de datos, en la mayoría de los programas gestores de bases de datos se denominan tablas. Dichas tablas son estructuras de filas y columnas que albergan datos referidos a un mismo tema. Cada fila, llamada ahora registro, contiene la información que antes estaba plasmada en una ficha del fichero. Cada columna de una tabla representa un campo. En la celda de la tabla en la que intersecta una fila con una columna tendremos un determinado campo dentro del cual normalmente se albergará un dato.

El siguiente paso es confeccionar un censo de las informaciones o datos que se albergarán en cada tabla: Definición de campos para las tablas de la aplicación; definir las informaciones necesarias y de utilidad para la gestión que cada tabla deberá albergar; definir los campos que va a tener cada tabla, así como su tipo y propiedades más indicadas (siendo conscientes de que las propiedades de los campos se van a poder modificar a posteriori). A este proceso se le reconoce también, de forma más técnica, como Normalización de Datos.

Aplicación en Rough Sets. Es claro que el proceso descrito anteriormente es de fácil aplicación, las tablas de información requeridas por la TRS, es mas, es necesario implementar las técnicas de normalización sobre un modelo de datos definido, según el ámbito del problema a estudiar. Que para el caso de ejemplo que se está teniendo en cuenta se pueden identificar entidades como: Docente, Nivel de estudios, Curso, Evaluación, Investigación, entidades que tienen sus propios atributos, dentro de un dominio específico; pero que al llevarlos al esquema de la TRS se normalizan, de tal manera que permitan visualizar de forma práctica el comportamiento de algunos de estos atributos frente a otros, en una sola representación de la información, llamada Tabla de Información.

Tabla 5. Ejemplo de la matriz de Información en TRS

Docentes	Nivel de estudios	N.° de cursos	Nivel de evaluación	Nivel de investigación	Buen docente
Docente-1	Pregrado	4	Bajo	Bajo	No
Docente-2	Maestría	3	Normal	Alta	Si
Docente-3	Pregrado	4	Normal	Bajo	Si
Docente-4	Doctorado	2	Alta	Muy alta	Si
Docente-5	Maestría	3	Muy alta	Muy alta	Si

Referencias

[1] Batini, C.; Ceri, S.; Navathe, S. Diseño Conceptual de Bases de Datos: Un enfoque de entidades interrelaciones, Addison-Wesley/Díaz de Santos, 1994.

[2] Date, C. J. Introducción a los sistemas de bases de datos, Vol. I (5.ª edición) Addison Wesley Iberoamericana, 1993.

[3] Celma, M.; Casamayor, J.C.; Mota, L. Bases de Datos Relacionales. Pearson – Prentice Hall, 2003.

[4] Elmasri, R.; Navathe, S. Fundamentals of database systems. Benjamin Cummings, 1994.

[5] Rumbaugh, Jacobson, Booch. The Unified Modeling Language Reference Manual, Addison Wesley 1998.

[6] Jacobson, Booch, Rumbaugh. The Unified Software Development Process, Addison Wesley 1999.

[7] Marakas, G. Decision Support Systems in the 21st Century, Prentice-Hall, New York, E.U.A. 1998.

[8] Pawlak, Z. Rough Sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht/ Boston/London. 1991.

[9] Pawlak, Z.; Slowinski, R. Decision Analysis using Rough Sets. ICS Research Report no. 21, Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland. 1993.

[10] Laudon, Kenneth C. Administración de los sistemas de información. 3ra. Edición. México. 1996. pp. 271-295.

[11] Rish, Irina. "An Empirical Study of the Naive Bayes Classifier". IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.

[12] Mitchell, T. Machine Learning, McGraw Hill, NY, 1997.

[13] Liu H. and H. Motoda. Feature Selection for Knowlegde Discovery and Data Mining. Kluwer Academic Publishers, London, UK, 1998.

Fecha de recepción: 20 de febrero de 2007
Fecha de aprobación: 14 de agosto de 2007