Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 347-352 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

DESARROLLO DE UN COMPONENTE DE SOFTWARE PARA EL ANALISIS DE DATOS DE UN SISTEMA SMART HOME

Jaime González Gordillo, Ernesto Eduardo Quiroz Morones, José Cruz Núñez Pérez
Centro de Investigación y Desarrollo de Tecnología Digital, Instituto Politécnico Nacional (CITEDI-IPN)
Av. del Parque 1310, Mesa de Otay, Tijuana B.C., C.P. 22510. Tel: (664) 6231344, FAX: (664) 6231344
jglezgordillo@gmail.com, eequiroz@citedi.mx, nunez@citedi.mx

RESUMEN

Una casa inteligente (en ingles *Smart Home*) está integrada por un sistema automático avanzado "inteligente", la cual puede controlar muchos aspectos de la vida cotidiana. La interacción entre el usuario y el sistema debe ser lo más natural posible, dando como resultado un sentimiento de aceptación por parte del usuario. Para que exista esta interacción natural, el sistema debe de adaptarse a las costumbres del usuario, por lo que el sistema debe recopilar datos del entorno, analizarlos y tomar una decisión la cual se transforma en un servicio hacia el usuario.

La toma de decisión que debe realizar el sistema requiere un análisis previo de los datos obtenidos; este análisis de datos debe hacer uso de la minería de datos que proporciona diversos métodos de análisis.

En presente artículo describe el desarrollo de un componente de software que permite el análisis de datos de un sistema Smart Home. Este componente es Open Source desarrollado en Java el cual puede ser integrado fácilmente en sistemas Smart Home que se adapten a los requerimientos del componente.

1. INTRODUCCIÓN

En la actualidad es posible almacenar gran cantidad de información de diversas índoles en bases de datos. Esta información es recogida de diferentes ámbitos de la sociedad y su tamaño crece día con día. Sin embargo, el procesamiento de esta información y el descubrimiento de conocimiento de este enorme volumen de datos es un reto actual [1]. Como resultado la minería de datos DM (por sus siglas en ingles *Data Mining*) es una nueva disciplina que está orientada al descubrimiento de conocimiento de una base de datos mediante métodos avanzados de estadística e inteligencia artificial [2]. La minería de datos está relacionada con diversas disciplinas como la base de datos, computación paralela, estadística, la inteligencia artificial entre otros [3, 4]. La disciplina DM ayuda a la extracción de información útil y no evidente de grandes bases de datos lo que permite a diversos ámbito a focalizar sus esfuerzos alrededor de la información importante contenida en su almacén de datos (en ingles *data warehouses*) [5].

Este artículo centra su investigación en los sistemas Smart Homes debido a que en estos sistemas se registran una gran cantidad de datos que son obtenidos a través sensores y actuadores que integran el sistema. Debido a lo anterior se ha desarrollado un componente de software que realice el análisis de datos y proporciones datos valiosos para facilitar la toma de decisiones en los sistemas Smart Homes.

2. DESARROLLO

El análisis de datos es un subconjunto de lo que actualmente se conoce como "descubrimiento de conocimiento en base de datos" (en ingles *Knowledge Discovery in Databases*, KDD). El proceso KDD involucra una serie de etapas que permiten el desarrollo del proceso de una manera eficiente. Esto conlleva a aplicar diferentes métodos en cada etapa lo que le adhiere complejidad al proceso KDD. El conjunto de etapas que involucra el proceso KDD se puede apreciar en la figura 1.

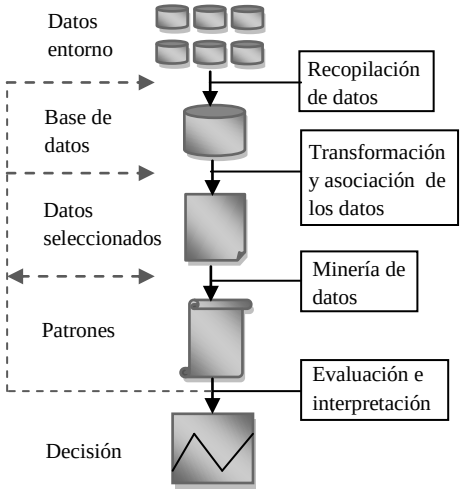

Figura 1. Etapas del proceso de descubrimiento de conocimiento en base de datos

Por otro lado el análisis de datos es el "analisis (inteligente) de datos para la extracción de

patrones" que hace mayor hincapié en las técnicas de análisis estadístico.
El objetivo de esta investigación es desarrollar un componente que brinde información importante para la toma de decisión. En este contexto es necesario realizar las etapas anteriores a la minería de datos para una mayor eficiencia en el proceso de análisis.
El componente de software desarrollado efectúa un conjunto de etapas para el análisis de datos las cuales son:

- Recopilación de datos
- Transformación y asociación de datos
- Minería de datos

Cada etapa realiza un conjunto de procesos que integran algoritmos computacionales que permiten el desarrollo de la misma.

2.1. Recopilación de Datos

Para poder comenzar a analizar y extraer información de los datos, primeramente debe de recopilarse los datos. Los datos deben ser almacenados en una sola tabla de la base de datos del sistema con el siguiente formato:

Tabla 1. Formato de tabla.

clave_objeto	a	m	d	h	mi	estado_objeto

donde:
clave_objeto representa un sensor o actuador.
a=año, $a = \{2012,2013,2014, \dots\}$
m=mes, $m = \{1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12\}$
d=día, $d = \{1, 2, 3, \dots ,31\}$
h=hora, $h = \{0,1, 2, \dots ,23\}$
mi=minuto, $mi = \{0,1, 2, \dots ,59\}$
estado_objeto $= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11\}$.

Se considera que todos los objetos deben tener mínimamente un evento registrado cada cinco minutos, este requerimiento es necesario para evitar la ausencia de datos en el proceso de análisis.
El análisis de datos se realiza únicamente con la tabla descrita anteriormente. El usuario debe proporcionar los siguientes datos:

- Usuario de MySQL
- Contraseña de MySQL
- Nombre de la base de datos
- Nombre de la tabla
- Conocimiento deseado

Con los datos proporcionados y la tabla descrita se podrá realizar la recopilación de datos para cada proceso a realizar.

Se realiza una copia de la tabla en la base de datos del sistema para evitar posibles problemas al realizar un nuevo proceso para la actualización de valores.

2.2. Transformación y asociación de los datos

En la etapa de transformación y asociación se efectúan varios procesos que involucran diversos algoritmos computacionales. Estos procesos deben ser realizados para que los datos estén en condiciones para su análisis en la etapa de minería de datos. La figura 2 muestra los procesos a realizar en esta etapa:

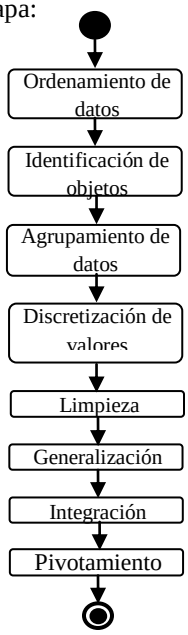

Figura 2. Etapas de la transformación y asociación de los datos.

2.2.1. Ordenamiento de datos

Se realiza un ordenamiento de los datos con el siguiente orden:

- idObjeto ,año, mes, dia, hora, minuto

El ordenamiento de datos permite separar los registros de cada objeto con el fin de poder evaluarlos en los siguientes procesos.

Tabla 2. Datos ordenados después del proceso de ordenamiento.

Registros del objeto 1

clave_objeto	a	m	d	h	mi	estado_objeto
A1	2012	1	1	4	3	7
A1	2012	1	1	4	4	2
B1	2012	1	1	4	1	9
B1	2012	1	1	4	4	3

Registros del objeto 2

2.2.2. Identificación de objetos

Se identifican los datos diferentes del atributo clave_objeto, esto permite conocer los diferentes

objetos (sensores y actuadores) que integran el sistema.

Tabla 3. Identificación de objetos en el atributo clave_objeto

clave_objeto	a	m	d	h	mi	estado_objeto
A1	2012	1	1	4	3	7
A1	2012	1	1	4	4	2
B1	2012	1	1	4	1	9
B1	2012	1	1	4	4	3

Se guada los datos diferentes en una matriz:
int idObjeto[]={A1, B1}
Esta matriz es utilizada en los siguientes procesos.

2.2.3. Agrupamiento de datos

Se realiza un agrupamiento de datos tomando de referencia el atributo mi, donde

$$mi = \{0,1,2,\dots,59\}$$

El agrupamiento se realiza con intervalos iguales, siendo el intervalo de 5 valores por agrupamiento.
El agrupamiento se realiza por cada objeto diferente (se hace uso de la matriz idObjeto).
Los intervalos de cada agrupamiento son los siguientes:

Tabla 4. Intervalos de agrupamiento.

$0 \leq m0 < 5$	$30 \leq m6 < 35$
$5 \leq m1 < 10$	$35 \leq m7 < 40$
$10 \leq m2 < 15$	$40 \leq m8 < 45$
$15 \leq m3 < 20$	$45 \leq m9 < 50$
$20 \leq m4 < 25$	$50 \leq m10 < 55$
$25 \leq m5 < 30$	$55 \leq m11 < 60$

Tabla 5. Datos agrupados.

clave_objeto	a	m	d	h	mi ($m0$)	estado_objeto
A1	2012	1	1	4	3	7
A1	2012	1	1	4	4	2
B1	2012	1	1	4	1	9
B1	2012	1	1	4	4	3

($m0$)

El agrupamiento es el paso anterior para el proceso de discretización de valores. Con este agrupamiento los datos estarán ordenados para su selección y actualización de valores.

2.2.4. Discretización de valores

La discretización o cuantización es la conversión de un valor numérico en un valor nominal ordenado que representa un intervalo.
A partir del agrupamiento realizado en el proceso anterior, se realiza una discretización de los valores del atributo mi y estado_objeto de cada agrupamiento. Para la discretización del atributo estado_objeto debe obtenerse el promedio de los elementos de cada agrupamiento. Posteriormente se realiza una comparación del valor promedio con los valores de la tabla 6 para obtener el valor nominal. Para la discretización del atributo mi, se realiza la comparación con los valores nominales de los intervalos correspondientes de la tabla 5.
En la siguiente tabla se presenta los valores nominales:

Tabla 6. Discretización de valores.

Valor numérico	Valor nominal
9-11	Muy alto
6-8	Alto
3-5	Medio
0-2	Bajo

Tabla 7. Extracción de valores referenciados por su agrupamiento.

clave_objeto	a	m	d	h	mi ($m0$)	estado_objeto
A1	2012	1	1	4	3	7
A1	2012	1	1	4	4	2
B1	2012	1	1	4	1	9
B1	2012	1	1	4	4	3

($m0$)

$$P_{X_0} = \frac{7+2}{2} = 4.5$$ Valor nominal = Medio

$$P_{X_0} = \frac{9+3}{2} = 5.5$$ Valor nominal = Medio

Posteriormente se actualiza los valores de los elementos de los atributos mi y estado_objeto de cada agrupamiento.

Tabla 8. Datos discretiados.

clave_objeto	a	m	d	h	mi	estado_objeto
A1	2012	1	1	4	m0	Medio
A1	2012	1	1	4	m0	Medio
B1	2012	1	1	4	m0	Medio
B1	2012	1	1	4	m0	Medio

2.2.5. Limpieza

Se realiza una limpieza de datos y atributos. Se eliminan los datos que estén en el mismo grupo dejando un único dato representativo del grupo. Se elimina los atributos a y m, debido a la irrelevancia que tienen en los procesos posteriores.

Tabla 9. Limpieza de datos y atributos.

clave_objeto	d	h	mi	estado_objeto
A1	1	4	m0	Medio
B1	1	4	m0	Medio

2.2.6. Generalización

La generalización consiste en sustituir valores generalizándolo de manera que sea más entendible y útil para el análisis. Los atributos d, h y mi se integraran en el proceso siguiente, por lo que es necesario realizar una generalización de los valores de los atributos d y h para su comprensión después de integrarlos.

Los valores del atributo d se consideran que su ordenamiento no afecta algún proceso posterior por lo que todos los valores son generalizados con un valor nominal consecutivo:

D = {d1, d2, d3,…,dn}

La generalización de los valores del atributo h si afecta su ordenamiento por lo que se debe considerar. La generalización de los valores del atributo h sólo se antepondrá la letra h en cada valor.

H = {h1, h2, h3,…,h23}

Tabla 10. Generalización de datos.

clave_objeto	d	h	mi	estado_objeto
A1	d1	h4	m0	Medio
B1	d1	h4	m0	Medio

2.2.7. Integración

La integración consiste en conseguir datos del mismo objeto que se puedan unificar.

En este proceso se integra los atributos d, h y mi para representarlo como un solo atributo.

Tabla 11. Integración de datos.

clave_objeto	tiempo	estado_objeto
A1	d1h4m0	Medio
B1	d1h4m0	Medio

2.2.8. Pivotamiento

El pivotamiento es el cambio de filas por columnas con el objetivo de tener una representación de atributo-valor. Esto permite tener una visualización más clara de las relaciones entre los datos. Debe considerarse el atributo tiempo, con el fin de representar los valores nominales que integran el mismo tiempo.

Tabla 12. Pivotamiento de datos.

Instante	A1	B1
d1h4m0	Medio	Medio

Se puede apreciar que el contenido de las tablas disminuyó considerablemente, solo se representan los datos que aportan información relevante.

Al finalizar todos los procesos anteriores, los datos están en óptimas condiciones para su análisis en la etapa de minería de datos.

2.3. Minería de Datos

La minería de datos puede definirse como un proceso iterativo de extracción de patrones predictivos ocultos en grandes bases de datos, utilizando las tecnologías de la inteligencia artificial, así como técnicas estadísticas [1].

En esta investigación se utilizó clasificadores Bayesianas una técnica estadística la cual se basa en el teorema de Bayes. Los clasificadores Bayesianos han demostrado una alta exactitud y velocidad cuando se han aplicado a grandes bases de datos. Estudios recientes han demostrado que el clasificador "Naive Bayesiano" es comparable en rendimiento a un árbol de decisión y a clasificadores de redes neuronales. A continuación se explica los fundamentos de los clasificadores bayesianos.

Se desea saber cuál es la mejor hipótesis dados datos históricos. Si denotamos P(D) como la probabilidad a priori de los datos, P(D|h) la probabilidad de los datos dada una hipótesis, y que queremos estimar es: P(h|D), la probabilidad a posteriori de h dados los datos históricos. Esto se puede estimar con el teorema de Bayes:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \qquad \text{Ec. 1}$$

Para estimar la hipótesis más probable (MAP) se busca el mayor P(h|D):

$$h_{MAP} = argmax_{h\epsilon H} P(h|D) \qquad \text{Ec. 2}$$

$$= argmax_{h\epsilon H} \frac{P(D|h)P(h)}{P(D)} \qquad \text{Ec. 3}$$

$$= argmax_{h\epsilon H} P(D|h)P(h) \qquad \text{Ec. 4}$$

Debido a que P(D) es una constante independiente de h.

Si se asume que todas las hipótesis son igualmente probables, entonces resulta la hipótesis de máxima verosimilitud (ML) es:

$$h_{ML} = argmax_{h\epsilon H} P(D|h) \qquad \text{Ec. 5}$$

El clasificador Naive Bayesiano se utiliza cuando se quiere clasificar una situación específica que involucra un conjunto de atributos a_i en un conjunto finito de clases (V).

$$V_{MAP} = argmax_{v_j \epsilon V} P(v_j | a_1, \dots, a_n) \quad \text{Ec. 6}$$

$$V_{MAP} = argmax_{v_j \epsilon V} \frac{P(a_1, \dots, a_n | v_j) P(v_j)}{P(a_1, \dots, a_n)} \quad \text{Ec. 7}$$

$$V_{MAP} = argmax_{v_j \epsilon V} P(a_1, \dots, a_n | v_j) P(v_j) \quad \text{Ec. 8}$$

El clasificador Naive Bayesiano asume que los valores de los atributos son condicionalmente independientes dado el valor de la clase:

$$P(a_1, \dots, a_n | v_j) = \Pi_i P(a_i | v_i) \quad \text{Ec. 9}$$

$$P(a_1, \dots, a_n | v_j) = P(v_j) \times \Pi_i P(a_i | v_i) \quad \text{Ec. 10}$$

Esta suposición de independencia se llama "independencia condicional de clase". Ésta simplifica los cálculos involucrados y disminuye el procesamiento computacional.
El componente de software desarrollado realiza un análisis de datos mediante el clasificador Naive Bayesiano. Se presenta a continuación el proceso de análisis de datos del componente.

Tabla 13. Datos después del proceso de transformación y asociación de datos.

Instante	A1	B1	C1	D1
d1h6m0	Medio	Medio	Muy alto	Bajo
d2h6m0	Medio	Medio	Muy alto	Bajo
d3h6m0	Alto	Alto	Bajo	Muy alto
d4h6m0	Medio	Medio	Bajo	Alto
d4h6m0	Alto	Alto	Bajo	Alto
d4h6m0	Medio	Bajo	Bajo	Bajo

donde
A1-Sensor de temperatura
B1-Sensor de humedad
C1-Persianas
D1-Aire acondicionado

Se desea conocer:

Tabla 14. Conocimiento deseado.

A1	B1	C1	D1
Medio	Medio	Bajo	?

El análisis arroja los siguientes datos
Tabla 15. Probabilidades a priori de D1.

D1			
Bajo	Medio	Alto	Muy alto
3/6	0/6	2/6	1/6

Tabla 16. Probabilidades a priori de A1.

A1				
	Bajo	Medio	Alto	Muy alto
Muy alto	0/3	0	0/2	0/1
Alto	0/3	0	1/2	1/1
Medio	3/3	0	1/2	0/1
Bajo	0/3	0	0/2	0/1

Tabla 17. Probabilidades a priori de B1.

B1				
	Bajo	Medio	Alto	Muy alto
Muy alto	0/3	0	0/2	0/1
Alto	0/3	0	1/2	1/1
Medio	2/3	0	1/2	0/1
Bajo	1/3	0	0/2	0/1

Tabla 18. Probabilidad a priori de C1.

C1				
	Bajo	Medio	Alto	Muy alto
Muy alto	2/3	0	0/2	0/1
Alto	0/3	0	0/2	0/1
Medio	0/3	0	0/2	0/1
Bajo	1/3	0	2/2	1/1

Se realiza las probabilidades correspondientes:

$$P(Muy\ alto) \times \Pi_i P(a_i | \text{Muy alto}) = \frac{1}{6} \times \frac{0}{1} \times \frac{0}{1} \times \frac{1}{1} = 0$$

$$P(Alto) \times \Pi_i P(a_i | \text{Alto}) = \frac{2}{6} \times \frac{1}{2} \times \frac{1}{2} \times \frac{2}{2} = 0.0833$$

$$P(Medio) \times \Pi_i P(a_i | \text{Medio}) = 0$$

$$P(Bajo) \times \Pi_i P(a_i | \text{Bajo}) = \frac{3}{6} \times \frac{0}{3} \times \frac{1}{3} \times \frac{1}{3} = 0$$

Se elige la probabilidad más alta, por lo que la hipótesis más probable es la de 0.0833

Tabla 19. Resultado del conocimiento deseado.

A1	B1	C1	D1
Medio	Medio	Bajo	Alto

El análisis de datos descubre patrones el cual puede ayudar a predecir actividades futuras siendo un aspecto fundamental para los sistemas Smart Homes.

3. RESULTADOS

Se han realizado diversas pruebas de ejecución del componente de software desarrollado y los resultados de cada prueba han sido favorables.
De manera resumida las pruebas realizadas en el componente son las siguientes:

- Pruebas unitarias: Detectar errores en los datos, lógica y algoritmos. Se realizaron pruebas en cada parte del componente de manera independiente y los resultados fueron favorables.
- Pruebas de integración: Detectar errores de interfaz y relaciones entre componentes. El componente se integró en sistemas Smart Homes y su funcionamiento fue adecuado.
- Pruebas del sistema: Detectar fallas en el cubrimiento de los requerimientos. Se realizaron diversas pruebas con diversos datos aleatorios, posteriormente el componente realizó un análisis de los mismos y arrojó resultados adecuados en cada conocimiento deseado.

El comportamiento del componente en todas las pruebas realizadas ha sido favorable.
El tiempo estimado de cálculo y el total de datos a analizar depende principalmente de la potencia computacional y no del algoritmo.

4. CONCLUSIONES

Los sistemas Smart Homes necesitan tomar decisiones basados en los datos obtenidos de su entorno. Estas decisiones deben ir acompañados de un análisis de datos y una debida evaluación para que la toma de decisión sea la correcta. Es por ello la gran importancia de desarrollar un componente que permita realizar tal análisis permitiendo facilitar la toma de decisiones a estos sistemas Smart Homes.
El presente trabajo aborda un componente de software desarrollado que brinda el servicio de análisis de datos para un sistema Smart Home. La principal aportación del presente trabajo reside en la maleabilidad que brinda el componente hacia sus usuarios, con la característica que es un componente Open Source el cual puede ser formado completamente al gusto de sus implementadores. El componente está diseñado de tal forma que se puede implementarse fácilmente en los sistemas Smart Homes.
Debido a los resultados favorables en la etapa de análisis se concluye que el componente de software presentado tiene un correcto funcionamiento al integrarse a sistemas Smart Homes y facilita la toma de decisiones de estos sistemas Smart Homes.
Entre los aspectos de trabajos futuros se encuentra la incrementación de análisis, esto es realizando el proceso de análisis de todos los posibles casos de una situación, con la finalidad de reportar al sistema las posibles relaciones que existan en los elementos involucrados de tal situación. Otro posible trabajo futuro es la paralelización automática convirtiendo el proceso de análisis del componente en un proceso multi-hilo, esto permitirá mejorar los tiempos de procesamiento de datos.

5. BIBLIOGRAFÍA

[1] José C. Riquelme, Roberto Ruiz, Karina Gilbert. Minería de datos: conceptos y tendencias. Revista Iberoamericana de Inteligencia Artificial. Vol. 10, Número 029. España.
[2] Salvador Torra. Del análisis estadístico a la minería de datos (data mining) mediante insightful miner. Universidad de Barcelona. España.
[3] Tomas Aluja. La minería de datos, entre la estadística y la inteligencia artificial. Universidad Politécnica de Catalunya. España.
[4] María N. Moreno García, Luis A. Miguel Quintales, Francisco J. García Peñalvo, M. José Polo Martín. Aplicación de técnicas de minería de datos en la construcción y validación de modelos predictivos y asociativos a partir de especificaciones de requisitos de software. Universidad de Salamanca. España.
[5] Luís Garrido, José Ignacio Latorre. Aplicaciones empresariales de data mining. Universidad de Barcelona. España.
[6] Ivica Crnkovic, Stig Larsson, Judith Stafford. Component-Based Software Engineering: Building systems from Components. At 9th IEEE Conference and Workshops on Engineering of Computer-Based Systems, New York, USA, 2002.
[7] V. Lakshmi Narasimhan, P. T. Parthasarathy, M. Das. Evaluation of a Suite of Metrics for Component Based Software Engineering (CBSE). Conference Issues in Informing Science and Information Technology. Volume 6, 2009.
[8] Romain Rouvoy, Philippe Merle. Leveraging component-based software engineering with Fraclet. Springer-Verlag France 2008.
[9] Arvinder Kaur, Kulvinder Singh Mann. Component Based Software Engineering. International Journal of Computer Applications. Volume 2 – No.1, May 2010.