

Comportamiento de Integración de Algoritmos para Descubrimiento de Reglas de Pertenencia a Grupos

José David Panchuk¹, Sebastian Martins², Horacio Kuna¹, Ramón García-Martínez²

1. Departamento de Informática. Facultad de Ciencias Exactas Químicas y Naturales.
Universidad Nacional de Misiones

2. Laboratorio de Investigación y Desarrollo en Ingeniería de Explotación de Información.
Grupo de Investigación en Sistemas de Información. Departamento Desarrollo Productivo y
Tecnológico. Universidad Nacional de Lanús
josepanchuk@gmail.com, smartins089@gmail.com, hdkuna@gmail.com,
rgm1960@yahoo.com

Resumen. Los procesos de explotación de información utilizan distintos algoritmos de minería de datos para obtener patrones de conocimiento a partir de los datos que se tienen sobre el dominio de problema. Uno de los supuestos con los que se trabajan estos algoritmos es que la complejidad del dominio de pertenencia de los casos que utilizan, no incide en la calidad de los resultados obtenidos. Es de interés analizar el comportamiento del proceso de explotación de información: Descubrimiento de Reglas de Pertenencia a Grupos, el cual utiliza algoritmos de clustering y algoritmos de inducción. En este trabajo se caracteriza la complejidad de los dominios en términos de las piezas de conocimiento que los describen y que los procesos de explotación de información buscan descubrir. Se muestra experimentalmente que en el caso del proceso de descubrimiento de reglas de pertenencia a grupos la calidad de los patrones que se obtiene difiere en función de los algoritmos que se utilizan en el proceso y de la complejidad de los dominios al cual aplican.

Palabras Clave. Ingeniería de Explotación de Información. Proceso de Explotación de Información. Complejidad de Dominios. Performance de par de algoritmos clustering e inducción.

1. Introducción

Varios autores [1-5] han señalado la necesidad de disponer de procesos de explotación de información que permitan obtener conocimiento a partir de las grandes masas de información disponible, su caracterización y tecnologías involucradas. En [6] se han definido cinco procesos de explotación de información: descubrimiento de reglas de comportamiento, descubrimiento de grupos, descubrimiento de atributos significativos, descubrimiento de reglas de pertenencia a grupos y ponderación de reglas de comportamiento o de pertenencia a grupos.

Los procesos de explotación de información utilizan distintos algoritmos de minería de datos para obtener patrones de conocimiento a partir de los ejemplos (instancias) que se tienen sobre el dominio de problema. Una de las hipótesis implícitas con las

que trabajan estos algoritmos es que fijados los algoritmos para el proceso de explotación de información, la complejidad del dominio sobre cuya información se aplican, no incide sobre la calidad de los patrones obtenidos. Sin embargo, hay indicios [7] que muestran que la complejidad de los dominios en términos de las piezas de conocimiento que los describen y que los procesos de explotación de información buscan descubrir, emerge como un componente no despreciable al momento de analizar la calidad de los resultados a obtener.

En este contexto, se busca demostrar mediante un experimento que para el caso del proceso de explotación de información descubrimiento de reglas de pertenencia a grupos, la calidad de los patrones producidos difieren en función de la complejidad de los dominios sobre los cuales se aplica y de los algoritmos que se utilizan en el proceso.

En suma, en este artículo se caracterizan los distintos tipos de complejidad de dominios y se expone el proceso de explotación de información descubrimiento de reglas de pertenencia a grupos (sección 2), se introduce el problema a partir del planteamiento de las preguntas de investigación (sección 3), se plantea el diseño experimental (sección 4), se describe el desarrollo del experimento (sección 5), y se presentan las conclusiones (sección 6).

2. Estado de la Cuestión

En esta sección se presentan los conceptos de proceso de explotación de información (sección 2.1) y de complejidad de dominios (sección 2.2).

2.1. Proceso de Explotación de Información: Descubrimiento de Reglas de Pertenencia a Grupos

En [6] se definen una serie de procesos de explotación de información los cuales guían al ingeniero de explotación de información en la identificación de los algoritmos de minería de datos a utilizar para la resolución de un problema de negocio particular. Para este artículo es de interés describir el proceso de descubrimiento de reglas de pertenencia a grupos el cual utiliza algoritmos de agrupamiento (clustering) y algoritmos de inducción para la extracción del conocimiento.

El proceso de descubrimiento de reglas de pertenencia a grupos aplica cuando el problema de negocio requiere identificar las características, descritas en términos de atributos y sus posibles valores, de un conjunto de clases a priori desconocidas, pero presente en la masa de información disponible sobre el dominio de problema. El proceso se puede describir de la siguiente manera: el primer paso consiste en identificar aquellas las fuentes de información existentes en la organización que posean datos relevantes para el problema en cuestión. Luego de identificadas las variables, estas se integran en una misma fuente de información (denominada datos integrados). Los algoritmos de agrupamiento se aplican una vez integrado los datos, obteniendo como resultado una partición del conjunto de registros en distintos grupos a los que se llama grupos identificados, generando posteriormente archivos asociados a cada grupo identificado (denominados como grupos ordenados). El atributo grupo de cada grupo ordenado se utiliza como el atributo clase de dicho grupo,

constituyéndose este en un archivo con atributo clase identificado (GR). Por último se implementa un algoritmo de inducción a partir del cual se obtienen un conjunto de reglas que definen el comportamiento de cada grupo.

2.2. Clasificación de Dominios por Complejidad

Para abordar el tema de la complejidad de los dominios, en [8] se propone caracterizar a los mismos en términos de piezas de conocimiento (reglas) que explican la pertenencia de una determinada instancia (ejemplo) a un determinado dominio. Es así que la complejidad del dominio queda caracterizada por la cantidad de clases que lo describe, la cantidad de reglas que definen la pertenencia a cada clase, la cantidad de atributos que puede tener cada regla y la cantidad de valores (distintos) que puede tener cada atributo.

Con base en los atributos de clasificación enunciados y el protocolo de clasificación de dominios propuesto en [7] se pueden clasificar en función de la complejidad de estos en los siguientes tipos:

“Dominios de Complejidad Simple: son aquellos dominios en que el aumento de la cantidad de ejemplos por regla, mejora el cubrimiento de reglas independientemente de las demás dimensiones utilizadas.

Dominios de Complejidad Mediana: son aquellos dominios que se explican con ejemplos con pocos atributos y pocas clases, ó pocos atributos y muchas clases ó pocas clases y pocas reglas por clase.

Dominios Oscilantes: son aquellos dominios que se explican con ejemplos donde pueden variar el número de atributos por ejemplo, ó cantidad de ejemplos soportados por una regla, o valores comunes de atributos en un conjunto de ejemplos cubiertos por la misma regla.

Dominios Complejos: son aquellos dominios que se explican con ejemplos con pocos atributos y muchos valores posibles por atributo, ó con muchos atributos y pocos valores posibles por atributo, ó con muchos atributos y muchos valores posibles por atributo.

Dominios Hipercomplejos: son aquellos dominios que se explican con ejemplos donde pueden variar la cantidad de posibles valores que pueden tomar los atributos, el número de atributos que cubren ejemplos, la cantidad de las reglas que cubren ejemplos, ó la cantidad de clases”.

3. Preguntas de Investigación

En este proyecto se han planteado las siguientes preguntas de investigación:

- ¿Es correcta la suposición que la performance de un par cualesquiera de algoritmos utilizados para el desarrollo del proceso de descubrimiento de reglas de pertenencia a grupos es independiente de la complejidad del dominio?
- En caso de falseamiento de la suposición previa: ¿Cuál es el par <algoritmo de agrupamiento, algoritmo de inducción> que brinda una mejor comprensión de la situación según la complejidad del dominio?

4. Diseño Experimental

Para el diseño experimental se ha desarrollado una herramienta, basándose en el banco de pruebas propuesto en [9]. Dicha herramienta fue desarrollada en lenguaje JAVA 1.8 y como gestor de base de datos PostgreSQL 9.3. Además, la herramienta hace uso de distintas librerías para la ejecución de los algoritmos de agrupamiento e inducción, estas son: WEKA 3.6 y 3.7, WEKA Classification Algorithms 1.8, KEEL, y otras librerías propias de JAVA.

La herramienta permite establecer la complejidad del dominio sobre el cual realizar la experimentación, para ello se deben definir de manera manual (o el sistema posibilita la asignación automática) de los parámetros del dominio relevantes, generando las variables libres (sin restricciones en sus valores) al azar. Los parámetros fundamentales a definir para la determinación de la complejidad del dominio son:

Cantidad de clases (CC): la cantidad de clases diferentes que creará la herramienta, cada clase contará con un conjunto de reglas propias y por consiguiente un conjunto de ejemplos, la cantidad de clases dictará también la cantidad de grupos en los que se agruparan la totalidad de los ejemplos generados en un experimento.

Cantidad de atributos (CA): la cantidad de atributos que tendrán los ejemplos a generarse, cada atributo a su vez tendrá una cierta cantidad de valores posibles que podrá tomar.

Cantidad de valores posibles por atributo (CVA): la cantidad de valores diferentes que puede llegar a tomar un atributo, tanto en el momento de generar una regla que impondrá una condición sobre cierto atributo (en ese caso la cantidad de valores posibles estará acotada, como se verá más adelante) o al generar los ejemplos.

Porcentaje de valores posibles por atributo en reglas (PCVA): el porcentaje sobre la cantidad de valores posibles por atributo que se tendrán en cuenta para armar las reglas.

Porcentaje de ejemplos usados para el entrenamiento de los algoritmos (PENT): el porcentaje de los ejemplos generados que se utilizarán para entrenar los algoritmos de agrupamiento e inducción antes de ejecutarlos sobre la totalidad de los mismos, los ejemplos de entrenamiento son seleccionados aleatoriamente a partir del conjunto de ejemplos completo.

Cantidad de reglas a generar por cada clase (CRC): la cantidad de reglas que se generarán para cada clase, las reglas generadas describen a cada clase, ya que a partir de estas se generan los ejemplos del dominio.

Cantidad de ejemplos a generar por cada regla (CER): la cantidad de ejemplos a generarse para cada regla, debe tenerse en cuenta que la cantidad de ejemplos dependerá de la cantidad de clases que se hayan definidos, de la cantidad de reglas a generar por clase que se haya definido y a este parámetro en si, por lo tanto la cantidad de ejemplos generados se dispara rápidamente dependiendo de los valores dados a los parámetros nombrados.

Cantidad de atributos a utilizar en cada regla (CAR): este parámetro puede traducirse a la cantidad de condiciones que tendrá cada regla generada por la herramienta, el valor de este parámetro deberá ser menor o igual a la cantidad de atributos definidos.

Semilla de generación de números aleatorios (SEM): la semilla que utilizará la herramienta para generar los números aleatorios que alimentan a todo el proceso de experimentación.

En la figura 1 se muestra la sección de carga de datos del dominio a generar de la herramienta, en la cual se observan los parámetros a configurar y la complejidad del dominio seleccionada.

Complejidad: simple	
Cant. Clases: 2	Cant. Reglas por Clase: 4
Cant. Atributos: 5	Cant. Ejemplos por Regla: 10
Cant. Val. Pos. Atributo: 8	Cant. Atrib. Regla: 2
Porc. Val. Atrib. Regla: 80	Seed: 451278
Porc. Entrenamiento: 17	iteraciones: 1
Guardar en: .\TURAIDESARROLLO\RESULTADOS SISTEMA	...
Nombre: []	

Fig. 1. Carga de Parámetros/Dominio

La definición de los dominios en condiciones de laboratorio conlleva la generación de un conjunto de reglas “originales”, a partir de estas reglas se generarán ejemplos que den soporte a las mismas, cada regla tiene el siguiente formato “*if att1 = val1 and att2 = val2... and attn = valn then Cn*”. La cantidad de reglas a generar y la cantidad de condiciones de las reglas, así como la cantidad de clases, atributos y valores diferentes, se definen en base a los valores establecidos por los parámetros del experimento.

El experimento está conformado por 3 pasos (figura 2):

- [i] En el primero se define la complejidad del dominio que se desea analizar y se determinan los parámetros de los mismos, como resultado de este caso se generan las reglas del dominio y la cantidad de casos que respaldan dichas reglas.
- [ii] El segundo paso consiste en la ejecución de cada uno de los posibles pares de algoritmos de minería de datos <clustering ; clasificación>, obteniendo así el conjunto de reglas descubiertas.
- [iii] El tercer y último paso consiste en la comparación entre el conjunto de reglas de clasificación generadas en el primer paso y las reglas descubiertas en el segundo paso. El porcentaje de reglas descubiertas de forma correcta, define el éxito del experimento [9].

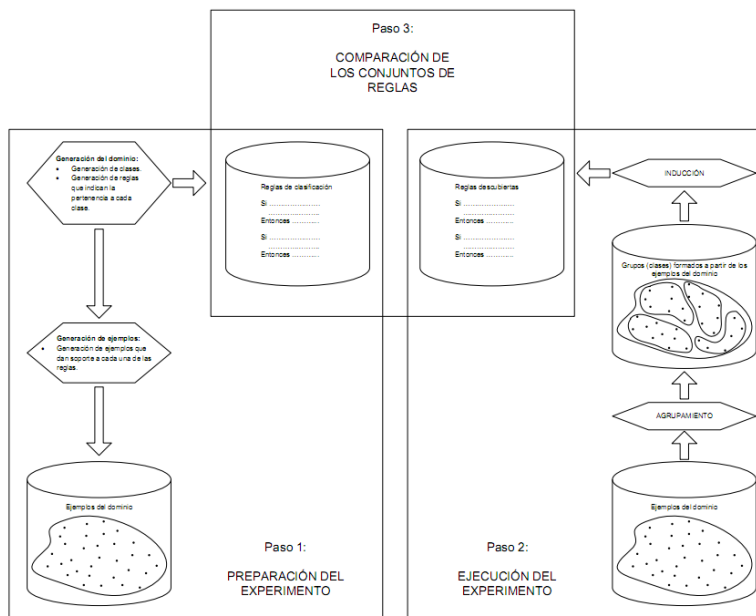


Fig. 2. Esquema de los pasos experimentales [9]

5. Desarrollo del Experimento

Para el desarrollo de la experimentación se realizaron 500 experiencias (100 escenarios para cada uno de los 5 tipos de dominios) a partir de las cuales se obtuvieron los resultados de análisis para cada una de las 30 combinaciones posibles de pares de algoritmos. Las variables de cada escenario fueron definidas de forma aleatoria según las variaciones posibles entre los dominios detallados en [7, 9]. Para la realización de los mismos se definieron categorías que establecen la variación de valores de cada variable de interés (tabla 1) y la relación entre dichas categorías (por parámetro) y la complejidad del dominio (tabla 2).

Tabla 1. Categorización de variación de Variables

VARIABLES	BAJO	MEDIO	ALTO
CC	2-4	5-7	8-10
CRC	1-5	6-10	11-15
CER	1-7	8-14	15-21
CAR	1-2	3-4	5-6
CVA	1-4	5-7	8-10
PCVA	1-40	41-70	71-100
CA	1-4	5-7	8-10

Tabla 2. Relación entre complejidad y variables independientes según el rango de variación

Complejidad\Variables	CC	CRC	CAR	CVA	CER	PCVA	CA
Simple					Alto		
Medianos	Bajo	Bajo	Bajo				Bajo-Medio
Oscilantes			Medio-Alto	Alto	Alto		
Complejos			Medio	Bajo-Medio	Alto	Medio	Medio
Hipercomplejos	Bajo	Bajo		Alto	Alto	Alto	Bajo

De la tabla 2 se deriva las posibles composiciones de cada escenario según su complejidad:

Escenarios de complejidad Simple: para los escenarios de complejidad simple la variable CER varía desde 15 hasta 21 mientras se mantiene fijo el valor del resto de las variables.

Escenarios de complejidad Media: en los escenarios de complejidad media, se varían las variables: Cantidad de Clases en el rango de 2 a 4, Cantidad de Reglas por Clase en el rango 1 a 5, Cantidad de Atributos Regla en el rango 1 a 2 y Cantidad de Atributos en el rango 1 a 5.

Escenarios de complejidad Oscilante: en los escenarios de complejidad Oscilante, se varían las variables: Cantidad de Atributos Regla en el rango 3 a 6, Cantidad de Valores Atributo en el rango 8 a 10 y Cantidad de Ejemplos por Regla en el rango 15 a 21.

Escenarios de complejidad Compleja: para los escenarios de complejidad Compleja, se varían las variables: Cantidad de Atributos Regla en el rango de 3 a 4, Cantidad de Valores Atributo en el rango 4 a 7, Cantidad de Ejemplos por Regla en el rango 15 a 21, Porcentaje de Valores Posible Atributo en el rango 45 a 70 y Cantidad de Atributos en el rango 5 a 7.

Escenarios de complejidad Hipercompleja: en los escenarios de complejidad Hipercompleja, se varían las variables: Cantidad de Clases en el rango de 2 a 4, Cantidad de Reglas por Clase en el rango 1 a 5, Cantidad de Valores Atributo en el rango 8 a 10, Cantidad de Ejemplos por Regla en el rango 15 a 21, Porcentaje de Valores Posibles Atributo en el rango 75 a 100 y Cantidad de Atributos en el rango 2 a 4.

De la ejecución de la experimentación, se generó una matriz por complejidad en la cual se detallan los resultados promedio de acierto de cada par de algoritmos a partir de los 100 casos generados. En las tablas 3 a 7 se presentan los resultados obtenidos.

En la tabla 3 se observan los resultados promedios obtenidos para dominios de complejidad Simple con cada combinación de algoritmos, se observa que la mejor combinación para este tipo de complejidad son los algoritmos SOM y ID3 con un 74.5% de reglas correctamente cubiertas en promedio, seguido por la combinación Farthest First y ID3 con un 66.75%, los resultados más altos desde el punto de vista inducción pueden verse en la columna del algoritmo ID3 y desde el punto de vista agrupamiento en la fila que pertenece al algoritmo SOM con sus respectivas combinaciones, los resultados más bajos se encuentran en las combinaciones de los algoritmos de agrupamiento con los algoritmos de inducción AQ15 y CN2

Tabla 3. Resultados Complejidad Simple

	PART	J48	ID3	A priori	CN2	AQ15
KMeans	50.5	57.25	60.37	51.37	15.25	8
EM	38.75	40.87	39.25	62.75	11.12	8
Farthest First	45.25	49	66.75	61.12	10.37	6.75
SOM	52.87	61	74.5	48.75	25.12	10.87
KNN	36.12	39	63.25	54.5	11.37	9.5

En la tabla 4 se encuentran los resultados promedios obtenidos para los dominios de complejidad Media usados, la mejor combinación de algoritmos para este tipo de complejidad fue Farthest First y ID3 con un 52.32% de reglas correctamente cubiertas en promedio, seguido por la combinación SOM y ID3 con un 51.70%, en general se puede observar que para este tipo de complejidad los resultados son menores en relación a los obtenidos para dominios de otras complejidades. Es interesante notar que las combinaciones de los algoritmos CN2 y AQ15 a pesar de seguir siendo los menores mejoraron en relación a los observados en la Tabla 5.5 de dominios Simples.

Tabla 4. Resultados Complejidad Media

	PART	J48	ID3	A priori	CN2	AQ15
KMeans	39.27	45.73	51.01	43.03	18.39	12.85
EM	32.84	34.97	41.72	49.63	15.18	9.71
Farthest First	35.05	38.11	52.32	46.08	15.70	8.94
SOM	40.61	45.16	51.70	40.03	18.09	11.26
KNN	38.66	41.45	45.74	34.10	18.76	13.01

En la tabla 5 se encuentran los resultados promedios obtenidos para los dominios de complejidad Oscilante, la mejor combinación de algoritmos para este tipo de complejidad fue KNN y A priori con un 94.25% de reglas correctamente cubiertas en promedio, seguido por la combinación EM y A priori con un 90.87%. Desde el punto de vista de los algoritmos de inducción claramente las combinaciones del algoritmo A priori presenta los resultados más altos, mientras que desde el punto de vista de los algoritmos de agrupamiento, el algoritmo KMeans con sus respectivas combinaciones tiene los promedios más altos, se observa que las combinaciones del algoritmo AQ15 fueron las más bajas y las combinaciones del algoritmo CN2 volvieron a aumentar.

Tabla 5. Resultados Complejidad Oscilante

	PART	J48	ID3	A priori	CN2	AQ15
KMeans	49.5	59.75	70.25	83.75	25.5	13.25
EM	39.62	48.5	48.75	90.87	21.87	11.5
Farthest First	48.75	53.5	71.5	83.75	22.37	9.25
SOM	29.62	30.87	61.62	84	20.12	2.5
KNN	9.62	9.37	19.5	94.25	13	1.5

En la tabla 6 se encuentran los resultados promedios obtenidos para los dominios de complejidad Compleja, la mejor combinación de algoritmos para este tipo de complejidad fue EM y A priori con un 90.25% de reglas correctamente cubiertas en promedio, seguido de la combinación KNN y A priori con 87%, nuevamente en este tipo de complejidad los resultados más altos se presentan en las combinaciones del algoritmo A priori con los algoritmos de agrupamiento, cabe destacar que en para este complejidad las combinaciones de ID3 también presentan valores altos y que desde el punto de los algoritmos de agrupamiento, las mejores combinaciones son aquellas donde esta presenta el algoritmo Farthest First.

Tabla 6. Resultados Complejidad Compleja

	PART	J48	ID3	A priori	CN2	AQ15
KMeans	50	53.37	68.87	81.75	19	12.37
EM	22.12	24.12	27.87	<u>90.25</u>	16.87	5
Farthest First	54.75	54.63	77.75	82.5	16.75	10.75
SOM	46.62	50	72.25	80.62	17.37	8.25
KNN	27.12	27.25	54.62	87	16.87	4.625

En la tabla 7 se observan los resultados promedios obtenidos para dominios de complejidad Hipercompleja con cada combinación de algoritmos, la mejor combinación de algoritmos para este tipo de complejidad fue **EM** y **A priori** con un 76.66% de reglas correctamente cubiertas en promedio, seguido de la combinación **Farthest First** y **A priori** con 70.84%, se observa además que en este tipo de complejidad se presentan los resultados más altos para las combinaciones con el algoritmo **CN2**, llegando a alcanzar y superar en ciertos casos a las combinaciones del algoritmo **PART**.

Tabla 7. Resultados Complejidad Hipercompleja

	PART	J48	ID3	A priori	CN2	AQ15
KMeans	36.74	41.60	49.21	66.86	37.04	18.24
EM	25.54	28.64	29.65	<u>76.66</u>	30.19	10.97
Farthest First	30.31	35.46	40.65	70.84	31.81	12.81
SOM	34.67	40.96	52.17	61.34	37.051	19.65
KNN	22.65	25.59	34.24	70.31	31.051	12.28

6. Conclusiones

A partir de los resultados previamente presentados, es posible aseverar que la performance alcanzada por los pares de algoritmos varía de acuerdo a la complejidad del dominio. Los mejores pares se presentan ordenados por la complejidad de manera creciente: <SOM, ID3>, <Farthest First; ID3>, <KNN, A PRIORI>, <EM, A PRIORI> y <EM, A PRIORI>. De los resultados obtenidos, se puede observar que el

algoritmo A PRIORI presenta mejoras significativas para las últimas 3 complejidades sin importar el algoritmo de clustering con el cual se lo empareje. Destacando que en las mismas obtuvo el máximo nivel de performance. El par <EM, A PRIORI> si bien presenta los mejores niveles de performance en las 2 últimas complejidades de dominio, en todas ellas presenta niveles aceptables de acierto. Los algoritmos PART y J48 presenta el mayor nivel de acierto general en el dominio más simple. Los algoritmos CN2 y AQ15 presentan los niveles más bajos de acierto, para todas las complejidades y combinaciones de algoritmos a nivel global. El algoritmo CN2 presenta mejoras en sus resultados a medida que la complejidad incrementa, si bien todos ellos están significativamente por debajo de la performance.

Como futuras líneas de investigación se prevé extender los algoritmos de agrupamiento e inducción en el software desarrollado y en una segunda etapa incorporar los procesos de explotación de información actualmente no considerados.

Financiamiento

Las investigaciones que se reportan en este artículo han sido financiadas por la beca PROMINF-UNLa-2015-2017 del Ministerio de Educación Argentina y parcialmente por el Proyecto UNLa-33A205 de la Secretaria de Ciencia y Tecnología de la Universidad Nacional de Lanús.

Referencias

- [1]Chen, M., Han, J., Yu, P. (1996). *Data Mining: An Overview from a Database Perspective*. IEEE Transactions on Knowledge and Data Engineering, 8(6): 866-883.
- [2]Chung, W., Chen, H., Nunamaker, J. (2005). *A Visual Framework for Knowledge Discovery on the Web: An Empirical Study of Business Intelligence Exploration*. Journal of Management Information Systems, 21(4): 57-84.
- [3]Chau, M., Shiu, B., Chan, I., Chen, H. (2007). *Redips: Backlink Search and Analysis on the Web for Business Intelligence Analysis*. Journal of the American Society for Information Science and Technology, 58(3): 351-365.
- [4]Golfarelli, M., Rizzi, S., Cella, L. (2004). *Beyond data warehousing: what's next in business intelligence?*. Proceedings 7th ACM international workshop on Data warehousing and OLAP. Pág. 1-6.
- [5]Koubarakis, M., Plexousakis, D. (2000). *A Formal Model for Business Process Modeling and Design*. Lecture Notes in Computer Science, 1789: 142-156.
- [6]García-Martínez, R., Britos, P., Rodríguez, D. (2013). *Information Mining Processes Based on Intelligent Systems*. Lecture Notes on Artificial Intelligence, 7906: 402-410.
- [7]Lopez-Nocera, M., Pollo-Cattaneo, F., Britos, P., García-Martínez, R. (2011). *Un Protocolo de Caracterización Empírica de Dominios para Uso en Explotación de Información*. Proceedings XVII Congreso Argentino de Ciencias de la Computación. Pág. 1047-1055.
- [8]Rancan, C., Pesado, P., García-Martínez, R. (2010). *Issues in Rule Based Knowledge Discovering Process*. Advances and App. in Statistical Sciences Journal, 2(2): 303-314.
- [9]Kogan, A., Rancán, C., Britos, P., Pesado, P. y García Martínez, R. (2007). Algunos resultados experimentales de la integración de agrupamiento e inducción como método de descubrimiento de conocimiento. Proceedings IX WICC, Pág. 11-15. Universidad Nacional de la Patagonia San Juan Bosco, Trelew, Argentina.