Algoritmos Genéticos para la Extracción de Reglas de Predicción Interesantes Aplicadas al Posicionamiento de los Distintos Motores de Búsquedas

Sergio M. Gallego, Héctor Oscar Nigro, Sandra González Císaro

INTIA- Departamento de Computación y Sistemas
Facultad de Ciencias Exactas - UNICEN
Campus Universitario - Paraje Arroyo Seco s/n
B7001BBO Tandil, Buenos Aires, ARGENTINA
TEL: +54-2293-439680 – FAX: +54-2293-439681
e-mails: sergiomgallego@gmail.com, onigro@exa.unicen.edu.ar, sagonci@exa.unicen.edu.ar

Abstract

The aim of this work is to infer different strategies for positioning a Web page in search engines. The challenge is to explore the relationship between different elements that make up the page and the final ranking in search engines as criterion and domain of pages. The case of study proposes to learn of the variables that determine the positioning of a page between the first ten positions in some motors search. We use inductive learning techniques, specifically; a genetic algorithm applied to the extraction of prediction rules with global search algorithms. This algorithm contains modifications to the classics, starting uniform population, discretization USD (Unparametrized Supervised Discretization), which maximizes the global kindness of the intervals that it obtains, and natural codification.. We experiment with real data for the positioning in various search engines, in the area of optimizing the same or also known as SEO (Search Engine Optimization).

Keyword: Ranking Web, Inductive Sorters, Optimization of Resources Web, Web Positioning, Genetic Algorithms, Rules of Prediction, SEO.

Resumen

El objetivo del trabajo se centra en inferir distintas estrategias de posicionamiento de páginas Web en motores de búsqueda. El problema consiste en explorar la relación entre distintos elementos que componen la página y el posicionamiento final en los motores de búsqueda según el criterio y el dominio de las páginas. El caso de estudio propone aprender de las variables que determinan el posicionamiento de una página entre las primeras diez posiciones en algunos motores de búsqueda. Utilizamos técnicas de aprendizaje inductivo, concretamente, un algoritmo genético aplicado a la extracción de reglas de predicción con algoritmos de búsqueda global. Este algoritmo contiene modificaciones con respecto a los clásicos, población inicial uniforme, discretización USD (Unparametrized Supervised Discretization), el cual maximiza la bondad global de los intervalos que obtiene, y codificación natural. Experimentamos con datos reales correspondientes al posicionamiento en los motores de búsqueda, en el área de optimización de los mismos.

Palabras claves: Ranking Web, Clasificadores Inductivos, Optimización de Recursos Web, Posicionamiento Web, Algoritmos Genéticos, Reglas de Predicción.

1. INTRODUCCIÓN

La optimización para motores de búsqueda (***S**earch **E**ngine **O**ptimization-SEO)* consiste en: a) aplicar diversas técnicas destinadas a lograr que los buscadores sitúen una página web en una posición y categoría deseada dentro de su índice de resultados para determinados conceptos clave de búsqueda; y b) conjunto de técnicas de desarrollo web que tienen como objetivo mejorar la posición de un determinado sitio web en la lista de resultados de los motores de búsqueda [23].
SEO implica diversas actividades como son programación, diseño y creación de contenidos con el objetivo de que una página, dependiendo de una clave de búsqueda, pueda ser situada en lo más alto de los resultados que entregan los buscadores.
Este artículo se enfoca en predecir el posicionamiento de una página web para un determinado criterio de búsqueda. La meta principal consiste en que el buscador posicione la página en los primeros diez lugares de resultados. Se trata de automatizar el trabajo del SEO, ya que los mismos modifican páginas web para obtener este posicionamiento de forma manual basados en experiencias previas y con años de trabajo. Con esta propuesta obtenemos reglas, con cierto nivel de precisión, que nos dirán, que atributos u objetos hay que modificar para obtener un buen posicionamiento dentro de los distintos buscadores.
Se propone un Algoritmo Evolutivo [18, 19], mas precisamente un algoritmo genético (AG) que obtendrá reglas de predicción para inducir si una página para un determinado criterio va a estar dentro de los primeros 10 lugares del ranking de los buscadores, y lo que se debe modificar en las mismas para que pueda llegar a estar en estos primeros lugares. El algoritmo propuesto se basa en GA-NUGGETS [8] con varias modificaciones como lo son una codificación natural para los individuos, una distribución uniforme para generar la población inicial y un método de discretización inicial llamado Unparametrized Supervised Discretization (USD).
El resto del artículo esta organizado de la siguiente manera: en la sección Algoritmo Genético explicaremos en detalle como trabajan los algoritmos genéticos empleados, sus funciones de adaptabilidad, los operadores genéticos, la discretización de los atributos y la población inicial. En la sección Experimentos y resultados, describiremos el archivo y las variables utilizadas en las pruebas; como así también los resultados de los experimentos realizados. Para finalizar las conclusiones y referencias bibliográficas.

2. ALGORITMO GENETICO

El objetivo del trabajo es la aplicación de Algoritmos Genéticos [18] para la extracción de reglas de predicción interesantes utilizadas para el posicionamiento de las páginas Web en los distintos motores de búsquedas. Se utilizará un AG Clásico con tres modificaciones importantes y de alta relevancia para poder obtener buenos resultados; una de estas modificaciones es utilizar una población uniforme [15], (en vez de aleatoria) para vencer problemas de la búsqueda genética. A partir de una población uniforme se logrará un espectro de búsqueda más amplio para que el algoritmo se expanda de manera uniforme en su búsqueda por el espacio de soluciones.
Otra modificación es utilizar una variación de un método de discretización, llamado USD Unparametrized Supervised Discretization [12], que tiene por objetivo transformar los atributos continuos en discretos, este método se utiliza, ya que maximiza la bondad de los intervalos generados. La tercera modificación consiste en utilizar una codificación natural [4] para obtener mayor eficiencia, ahorro de espacio y evitar continuas conversiones a la hora de codificar distintos genes.
Se desarrolla una aplicación que tiene como meta codificar un AG para la extracción de reglas de predicción interesantes, con las características anteriormente mencionadas. En esta aplicación se podrá configurar distintos parámetros del algoritmo, como así también, decidir cual será la función de aptitud para evaluar las distintas reglas, que en una primera instancia se tendrá en cuenta la predicción y el interés. Los parámetros del algoritmo tienen singular importancia, ya que, la variación de los mismos tiene gran impacto sobre las reglas producidas.
La aplicación será utilizada para procesar el archivo que contiene los factores que toman en cuenta los buscadores a la hora de posicionar una página en una búsqueda. La salida de la aplicación nos dará un conjunto de reglas predictivas, asociando a cada regla con distintas medidas, como es la precisión, soporte e interés, entre otras; estas medidas nos darán una visión de que tan buenas son las mismas [13, 14, 17, 22].
Con los resultados que se obtengan se puede predecir qué factores se deberán modificar y por qué valor, para que la misma pueda llegar a posicionarse dentro de los primeros diez lugares para una consulta determinada. No solo se entregan reglas para predecir el posicionamiento de una pagina sino que se entrega un conjunto de reglas que ayudan a clasificar las mismas, a estas reglas las llamaremos reglas para la clasificación. Esto se realizará mediante otro algoritmo genético que tomara estadística del primero para clasificar las reglas de manera correcta, un algoritmo genético dentro de otro. A modo de ejemplo, para una instancia estadística de una página, el primer AG nos dará un conjunto de reglas (reglas de predicción), de las cuales puede surgir que cierta cantidad de reglas nos informen que la pagina va a estar dentro de los primeros 10 lugares mientras que otra cierta cantidad nos puede inferir que no estará dentro de esos lugares, por lo tanto las reglas del segundo AG nos ayudaran a clasificar dicho problema. En

resumen, se entregará un conjunto de reglas predicativas para inferir el posicionamiento y reglas de clasificación para utilizar o clasificar las primeras.

2.1. Función Fitness

La función de adaptabilidad utilizada es la siguiente, planteada en GA-NUGGETS [8, 20] y se toma tal cual fue enunciada por Freitas, ya que para inducir las reglas se tendrá en cuenta solo la precisión mientras que para promover las reglas para clasificar se tendrá en cuenta también lo interesante de una regla:

$$\text{Fitness} = \frac{w_1 \cdot \frac{AntInt + ConsInt}{2} + w_2 . PredAcc}{w1 + w2}$$

Figura 1: Función de adaptabilidad

Donde **W1 y W2** son pesos creados para el usuario, ya que el mismo puede darle mayor peso a **PredAcc** que a **AntInt y ConsInt**. Finalmente todos los componentes de ecuación son normalizados.

2.2. Operadores Genéticos

Como son varios los pares posibles de <atributo objetivo, valor de atributo objetivo> que puede ocurrir en el consecuente de la regla, en cada generación, la mejor regla que predice cada par de <atributo objetivo, valor de atributo objetivo> es aprobado intacto para la siguiente generación. Esto corresponde a usar un factor de elitismo ***k***, dónde ***k*** es el número de consecuentes bien definidos de las reglas que ocurren en la población actual.
Se utiliza el operador **crossover de dos puntos** para obtener el cruce entre un par de individuos, obteniendo los individuos mediante la estrategia de **selección por torneo**. El operador de **mutación** y **Operadores de Generalización y Especialización** que insertan y remueven genes, los cuales directamente tratan de controlar el tamaño de las reglas que están siendo desarrollados, así también se influencia la comprensibilidad de las reglas. Estos operadores insertan o remueven aleatoriamente una condición en el antecedente de regla.
Para evitar la Endogamia (término aplicado a ciertas costumbres que se practican en algunas sociedades, por las cuales un miembro de una comunidad, tribu, clan o unidad social contrae matrimonio con otra persona del mismo grupo social), se utiliza un **operador Endogámico** que inserta cromosomas nuevos o limpios a la población en cada generación, con este operador tratamos de forma directa una de las deficiencias de los AG, que es caer en un máximo local.
Los operadores genéticos previos actúan en el antecedente y el consecuente de la regla.
Después de que estos operadores genéticos son aplicados, el algoritmo analiza si fue creado algún individuo inválido. Si es así, se usa un **operador de reparación** para producir individuos con genotipo válido [18].

2.3. Método de discretización: Unparametrized Supervised Discretization (USD)

Un objetivo es la disminución de la cardinalidad de los atributos continuos de una base de datos, especialmente en procesos de minería de datos, donde muchos algoritmos de aprendizaje sólo operan en espacios finitos de atributos discretos o nominales [22]. Para trabajar con atributos continuos donde el rango de valores es infinito, puede ser recomendable la discretización de dichos atributos.
El **USD** fue desarrollado por Giradles, Aguilar y Riquelme en [11, 12] y el objetivo que persigue, es dividir los atributos continuos en intervalos de máxima bondad, de forma que la bondad media de todos los intervalos finales para un determinado atributo sea lo más alta posible, o sea, que cubran el máximo número posible de ejemplos sin que se produzca pérdida de bondad por parte de dichas reglas.

2.4. Población Inicial: Método Uniforme(UP)

Unos de los problemas importantes relacionados con AGs, es estar atrapado en la solución/soluciones local o desviarse de solución mejor o sub-mejor.
El método UP [15] implementa el método divide y conquista para generar a una población inicial de buena calidad. Con éste método, los cromosomas son distribuidos sobre espacio de cromosomas, entonces, encontrar en la población generada a la solución en la zona de influencia toma menos tiempo que población inicial aleatoria. A causa de cromosomas distribuidos sobre el espacio, es imposible estar atrapado en una solución local. Consecuentemente, la solución mejor o sub-mejor puede ser obtenida de este modo.

3. EXPERIMENTOS Y RESULTADOS

3.1. Parámetros de Entrada

Existe gran diversidad de los parámetros o atributos que tienen en cuenta los motores de búsquedas a la hora de posicionar una página Web en su ranking. En la tabla 1 se explicará brevemente algunos de los que utilizaremos para este proyecto, donde también se informará el formato y si el mismo se puede modificar en las páginas Web.

Nombre	Aclaración	Descripción	Formato	Modificable
CompetitonGoogle	Cantidad de resultados en Google para el criterio	Cantidad total de páginas que hacen matching con el criterio de búsqueda. El valor es estimado por google al realizar una búsqueda.	numérico – entero	No
DensityCriterio	Densidad del criterio	Corresponde al cociente entre la cantidad de ocurrencias de las palabras del criterio en la página relevante sobre la cantidad total de palabras en la página relevante	numérico %	Sí
ImportantCriterio	Importancia del Criterio	Este valor se calcula teniendo en cuenta la ubicación del criterio en la página (titulo, H1, H4, meta-keyword, meta-description, texto alt, contenido, url, dominio). Se realiza una suma ponderada de la densidad del criterio en cada parte de la página	numérico - real (2 decimales)	Sí
CalculatePR	Page Rank	El PR (PageRank) es un valor numérico asignado por Google a cada dominio que representa la importancia que una página web tiene en Internet. Los valores asignados por Google van de 0 a 10. El valor 0 representa la menor importancia y 10 la mayor importancia	numérico – entero	No
cmpTotalPagesIndexesGoogleCriterio	Cantidad de páginas indexadas por Google, que contienen el criterio	Los factores con cmp delante del nombre corresponden a los competidores. Actualmente se toman como datos de los competidores a los primeros 5 y se los promedia. Ningún factor de la competencia es modificable por obvias razones. "Los factores de los competidores se calculan promediando los valores obtenidos (siempre que sean válidos, no se promedian los -1 ni valores especiales, PR = 99	numérico – entero	No
……	…….	…….	……..	……..
Position	Posición en Google	Posición en el resultado de búsqueda de la página relevante al buscar el criterio. Si la posición esta entre los primeros 10 lugares, retorna 0, en caso contrario retorna 1.	numérico	Dato a inferir

Tabla 1 Algunas de las variables incluidas en el archivo de entrada

El archivo de entrada contiene los atributos e instancias recopiladas. Dependiendo de los atributos, las páginas para una determinada búsqueda de un criterio se posicionan en los primeros 10 resultados (Atributo "posición" = 0) y los que no (Atributo "posición" = 1). El archivo empleado contiene 4366 instancias con 24 atributos, de los cuales 23 son numéricos y 1 es nominal (la posición 0 o 1). Los atributos del segundo algoritmo son tomados de los resultados obtenidos del primer algoritmo, los promedios de las medidas confianza, precisión y soporte; los máximos de las mismas; las relaciones de las anteriores medidas; la cantidad de reglas producidas para la clase que clasifica, así como también la cantidad de genes que cubren en total las reglas para una determinada clase que clasifica. Este archivo se va incrementando a medida que se realiza la clasificación de modo que va aprendiendo como clasificar dependiendo de las clasificaciones anteriores.

3.2. Codificación

El enfoque utilizado para trabajar las reglas mediante algoritmos genéticos, es Michigan, o sea que un cromosoma es representado por una regla y no por un conjunto de ellas como lo es en el enfoque Pittsburg [9, 10].
Se plantea una codificación más compacta y en la que no hay que realizar constantes conversiones para la aplicación de los operadores genéticos. Para ello se tomo en cuenta la denominada Codificación Natural [1, 2, 4, 12], donde cada gen representa los valores de un atributo, ya sea continuo o discreto. Sin embargo, al contrario que en otras codificaciones, en la natural cada gen es un único número natural. Aunque los atributos continuos y discretos usen la misma codificación, el modo de codificar los valores de ambos tipos de atributos es diferente [5, 21].
La codificación Natural fue tomada con ciertas modificaciones. Para los atributos continuos, en un principio, solo se utiliza la diagonal de la matriz, ya que el uso de toda la matriz solapa los intervalos y se observa que se viola el principio de unicidad para las reglas de predicción.
En cuanto a los atributos discretos se opto por codificar cada valor y no un conjunto de valores como en la codificación natural clásica, los cuales representan una disyunción.
En este algoritmo se utilizará un genotipo de longitud variable que es equivalente al antecedente de la regla de longitud variable, o sea, nuestro cromosoma tendrá un tamaño variable dependiendo de los atributos que se encuentren o no en el antecedente. Se tendrá que tener especial cuidado en el surgimiento de cromosomas inválidos.
En cuanto al consecuente de las reglas, creemos que es más eficiente que evolucione con las reglas, ya que, de lo contrario se sesga el algoritmo y se pierde de vista el significado de la evolución natural. Esto es una de las principales diferencias con el algoritmo de Freitas [8], ya que el mismo utiliza una codificación binaria con cromosomas de longitud fija. Este autor propone seleccionar el mejor consecuente para un cromosoma en particular, mas específicamente, para cada cromosoma generado se escoge el mejor consecuente que maximice la función de adaptación. El algoritmo trabajando de esta manera se torna determinista, se estaría aproximando los AG a los algoritmos de inducción clásicos, en consecuencia el consecuente no evoluciona junto con la regla. Otra diferencia está en que la población no es fija, sino que evoluciona. Por lo tanto crece a medida que corren las generaciones, semejante a una población humana o animal.

Los resultados obtenidos son expresados mediante reglas de tipo predictivo:

TotalDMOZEntries ->[1.5,2.5] ; CalculatePR->[3.5,5.5] ; cmpCalculatePR->[2.775,3.63] → Posicion->0 - Fitness: 0.99 -Cubre: 53.0 reglas -Soporte: 0.012 -Confianza: 1.0 -Interesante: 0.38

La regla indica que si el total de entradas en el directorio DMOZ de nuestra pagina esta entre 1.5 y 2.5 y el Page Rank de nuestra pagina esta entre 3.5 y 5.5 y el promedio de Page rank de nuestro competidores esta entre 2.77 y 3.63 entonces nuestra pagina será posicionada dentro de los primeros 10 posiciones de los resultados entregados por Google. El *cmpCalculatePR* al ser un atributo no modificable, se puede utilizar para seleccionar la regla con la instancia que se provee como entrada, para luego llevar *TotalDMOZEntries* y *CalculatePR* del sitio a los valores que indica la regla.
Cada regla indica su Fitness, para este caso en particular la precisión; las instancias de entrenamiento que cubre; el soporte obtenido y la confianza. También informa que tan interesante es nuestra regla [3, 6, 7, 16].
Para trabajar con los archivos antes descriptos se discretizan los atributos numéricos. En el algoritmo genético se usa la discretización USD mientras que en Weka [24] se utiliza PKIDiscretize. En Weka hay que discretizar junto el archivo de entrenamiento y el de test, o sea, se discretiza solo un archivo y luego se divide en los anteriores. En el genético solo se discretiza el de entrenamiento mientras que el de test se ajusta a la discretización del archivo de entrenamiento. Si un gen no se encuentra dentro de tal discretización el mismo no es tomado en cuenta pero sí el resto de los atributos
El modelo generado por el AG esta compuesto por un conjunto de reglas con un mínimo de confianza y fitness, en muchas ocasiones no encuentran reglas para una determinada instancia porque no cumplen el mínimo de confianza y fitness requeridos. Se tiene en cuenta que el conjunto de reglas cubran todos los ejemplos, al menos una regla debe cubrir un ejemplo lo que asegura una máxima cobertura de las instancias de entrenamiento.

El segundo algoritmo trabaja con el fitness o función de ajuste de precisión e interés ya que los datos con los que se trabaja no están balanceados, hay mas instancias clasificadas bien que las clasificadas mal, por lo cual usar solo precisión no resulta útil ya que con este genético debemos dilucidar las reglas de clasificación que inducen mal y no las que clasifican bien. El fitness, desarrollado por Freitas [8]**,** necesita de dos parámetros w1 y w2 para esta función. Se ha optado por variarlo a lo largo de las folds.
El parámetro w1 es igual a 0.29 y w2=0.71 en un principio y luego van decreciendo y creciendo respectivamente a medida que se va incrementando la población para poder encontrar mas reglas que se han clasificado mal.
En tabla 2 observamos una comparación entre los resultados obtenidos por J48 y el AG propuesto, tomando en cuenta 10-Fold Cross Validation, el mismo archivo de entrenamiento y utilizando un archivo de testeo.

	J48		AG	
10-FCross-Validation (A)	Instancias Bien Clasificadas	3227	Instancias bien Clasificadas	2013
	Instancias Mal Clasificadas	1139	Instancias mal Clasificadas	408
	Precisión Modelo	73,91%	Precisión Modelo	83,14%
	Precisión de la Posición 0 (primeros 10 lugares)	72,8%	Precisión de la Posición 0(primeros 10 lugares)	92,7%
	Cobertura	100%	Cobertura	55,45%
Test train (4366 Instancias) (B)	Instancias Bien Clasificadas	4074	Instancias bien Clasificadas	3431
	Instancias Mal Clasificadas	292	Instancias mal Clasificadas	244
	Precisión Modelo	93,31%	Precisión Modelo	93,36%
	Precisión de la Posición 0 (primeros 10 lugares)	92,00%	Precisión de la Posición 0(primeros 10 lugares)	94,9%
	Cobertura	100%	Cobertura	84,17%
File test (Train 4000 – Test 366) (C)	Instancias Bien Clasificadas	253	Instancias bien Clasificadas	223
	Instancias Mal Clasificadas	113	Instancias mal Clasificadas	66
	Precisión Modelo	69,12%	Precisión Modelo	77,16%
	Precisión de la Posición 0 (primeros 10 lugares)	64,4%	Precisión de la Posición 0(primeros 10 lugares)	77,8%
	Cobertura	100%	Cobertura	78,96%

Tabla 2 Cuadro Comparativo entre J48 y Algoritmo Genético utilizando 10-Fold Cross Validation

En el apartado (A):

a) Encontramos que las instancias clasificadas correctamente son de 83,14%, o sea, 9,23% mas que la corrida del J48 basado en árboles.
b) La precisión de la clase 0, se nota una gran diferencia a favor del algoritmo propuesto, obtiene 92,7% contra un 72,8%, una ganancia respecto al J48 de 19,9%. Esta clase es la más importante del trabajo ya que es lo que se necesita para predecir los factores que hay que modificar para un posicionamiento entre los diez primeros lugares.

En el apartado (B):

Se observa que no hay gran diferencia entre los mismo ya que ambos modelos poseen una precisión alrededor de 93%. Si se observa una pequeña ventaja en la precisión con respecto a la clase 0 de un 2,9% a favor del AG.

En el apartado (C):

En este apartado se dilucida la misma tendencia que en el apartado (A), llevando la diferencia a 8.04% a favor del AG con respecto a la precisión del modelo y extendiendo a 13.4% la diferencia con respecto a la precisión de la clase 0.

Observadas las ganancias respecto del J48, el algoritmo genético propuesto tiene la desventaja de no tener la misma cobertura que el primero, para ser más específicos un 44,55% de instancias no clasificadas en el apartado (A). Este porcentaje se debe a varias razones:

a) Una que no se han encontrado reglas para las instancias en cuestión que cumplen con un minino de confianza y soporte, más específicos 73,62% de ese 44,55%; y otra debido al filtrado del segundo algoritmo 26,38% de 44,55%. Esto se observa en el cuadro comparativo 2, apartado(D).
b) De ese 26,38%, el 33,33% son errores reconocidos y 66,66% son instancias filtradas mal, apartado (D).
c) el 73,62% de las instancias no encontradas obedecen a que el algoritmo es computacionalmente costoso y en las corridas, para reducir los tiempos de entrega de los resultados, se han descuidado ciertos parámetros como por ejemplo la población inicial o el número de generaciones del mismo, ambas tendrían que ser mas elevadas para obtener una mayor cantidad de reglas. Igualmente se han tomado como confianza mínima 90% y precisión mínima 90%.

En la tabla 3, se demuestran las ganancias obtenidas por el 1° AG debido al 2° AG.
Cabe aclarar que, las instancias filtradas son aquellas que el AG cree que no debe clasificar; dentro de ellas se encuentran las que están bien y mal. Las filtradas bien corresponden a que si se hubiesen clasificado nos producirían errores, mientras que las filtradas mal nos hubiesen producido instancias clasificadas correctamente.
Las instancias NO filtradas son la cantidad de instancias que el 2° AG cree que se deben clasificar y por lo tanto deja que el 1° AG las prediga. A diferencia que las anteriores si las instancias son bien clasificadas las representa bien, mientras que si no lo son las representan mal.
Las instancias NO encontradas dilucidan las instancias que el 1° AG no pudo encontrar reglas para cubrir las mismas. Mientras que las NO clasificadas representan la totalidad de las instancias que fueron filtradas por el 2° AG

	Con Filtro del 2º AG			Sin Filtro del 2º AG		
10-FCross-Validation (D)	Instancias Filtradas	Bien	171	Instancias Filtradas	Bien	0
		Mal	342		Mal	0
	Instancias NO Filtradas	Bien	2013	Instancias NO Filtradas	Bien	2355
		Mal	408		Mal	579
	Instancias NO encontradas		1432	Instancias NO encontradas		1432
	Instancias NO clasificadas		513	Instancias NO clasificadas		0
	Precisión del Modelo		83,14%	Precisión del Modelo		80,26%
Test train (4366 Instancias) (E)	Instancias Filtradas	Bien	97	Instancias Filtradas	Bien	0
		Mal	594		Mal	0
	Instancias NO Filtradas	Bien	3431	Instancias NO Filtradas	Bien	4025
		Mal	244		Mal	341
	Instancias NO encontradas		0	Instancias NO encontradas		0
	Instancias NO clasificadas		691	Instancias NO clasificadas		0
	Precisión del Modelo		93,36	Precisión del Modelo		92,18%
File test (Train 4000 – Test 366) (F)	Instancias Filtradas	Bien	36	Instancias Filtradas	Bien	0
		Mal	41		Mal	0
	Instancias NO Filtradas	Bien	223	Instancias NO Filtradas	Bien	264
		Mal	66		Mal	102
	Instancias NO encontradas		0	Instancias NO encontradas		0
	Instancias NO clasificadas		77	Instancias NO clasificadas		0
	Precisión del Modelo		77,16%	Precisión del Modelo		72,13%

Tabla 3. Cuadro comparativo entre si es usado el 2° Algoritmo Genético o no lo es. El 2° AG proporciona reglas para clasificar las reglas del 1° AG.

Tanto en el apartado (D), (E) y (F) se dilucidan ganancias que aporta el 2° AG al 1°AG. Estas ganancias van desde 2,88%, 1,18% y 5,03% respectivamente. En conclusión se puede decir que la utilización del 2° AG realmente se justifica, tanto por aportar ganancia en la precisión del modelo como en el aporte de reglas de clasificación, ya que se nota una considerable mejora en el apartado (F) que simularía los futuros casos reales.

4. CONCLUSIONES

El AG propuesto obtiene reglas que no solo nos indican si una pagina va a estar en los primeros 10 resultados de una búsqueda sino que además nos indican que factores o atributos de nuestra pagina debemos modificar para obtener el objetivo principal. Además se incorpora la ayuda de las reglas de clasificación a la hora de decidir, lo cual no es trivial.
Por otro lado, observamos que en el 2º AG existe un TradeOff entre la ganancia que aporta el mismo y las instancias NO clasificadas, o sea, que a medida que aumenta la ganancia en la precisión aumenta las instancias NO clasificadas, ofreciendo menos cobertura que otros modelos.
Podemos afirmar que mas allá de la precisión que tenga el modelo obtenido, el cual a simple vista es superior a otros modelos, lo importante es la automatización del trabajo realizado por los profesionales SEO´s, ya que la labor de los mismos evoluciona con el pasar del tiempo y con la experiencia obtenida.
Actualmente se está trabajando con nuevas medidas asociadas con cada regla para obtener así conductas mas especificas de las mismas. A su vez se está mejorando la precisión del 2 AG que repercute de forma directa en la precisión del modelo definitivo, llevando la ganancia a mas del 5% en todos los ámbitos.
En los trabajos futuros se debe mejorar la eficiencia del algoritmo, ya que es muy costoso computacionalmente y no se puede expandir totalmente, por lo que no se obtienen la cantidad de reglas óptimas deseadas. Se debe tratar de

perfeccionar el método de discretización de modo que no se genere tanta cantidad de intervalos y en consecuencia el AG tarde en converger.
En cuanto a los algoritmos evolutivos se debe explorar si es más eficiente un AG que la programación genética. Para determinados problemas la programación genética seria muy ventajosa para expresar las interrelaciones que poseen los atributos.

REFERENCIAS

[1] J. S. Aguilar. *"Generación de Reglas Jerárquicas de Decisión con Algoritmos Evolutivos en Aprendizaje Supervisado"*. Tesis Doctoral, Univ. de Sevilla. 2001.

[2] Aguilar-Ruiz, J. S. & Riquelme, J. C. *"Improving the Evolutionary Coding for machina Learning Tasks"*. European Conference on Artificial Intelligence, ECAI '02 pp. 173-177, IOS Press, Lyon, France.

[3] D. R. Carvalho, A. A. Freitas & N. F. F. Ebecken. *"A Critical Review of Rule Surprisingness Measures"*. ICDM-Rio-2003.

[4] Giradles, R., Aguilar-Ruiz, J., Riquelme, J. y Mateos, D. *"Cogito*: Aprendizaje Evolutivo de reglas de Decisión con Codificación Natural*". Departamento de Leguajes y Sistemas Informáticos. Universidad de Sevilla.

[5] James Dougherty, Ron Kohavi & Mehran Sahami. *"Supervised and Unsupervised Discretization of Continuous Features"*. In Armand Prieditis & Stuart Russell. Eds, Machine Learning: Proccedings of the Twelfth International Conference, 1995, Morgan Kaufmann Publishers, San Francisco, CA.

[6] Freitas, A. A. *"On objective measures of rule surprisingness"*. Proc. 2nd European Symp. On Principles of Data Mining and Knowledge Discovery (PKDD-98). 1998.

[7] A. Freitas. *"On Rule Interestingness Measures"*. Knowledge-Based Systems Journal. Federal Center of Technological Education 1999.

[8] Freitas, A. A. *"A genetic algorithm for generalized rule induction"*. In: R. Roy et al. Advances in Soft Computing - Engineering Design and Manufacturing. 1999.

[9] A. Freitas. *"Understanding the crucial differences between classification and discovery of association rules"*. ACM SIGKDD Explorations. 2(1) pp. 65-69. 2000.

[10] A. Freitas. *"Data Mining and Knowledge Discovery with Evolutionary Algorithms"*. Springer-Verlag. 2002.

[11] Giradles, R., Aguilar-Ruiz, J., Riquelme, J. *"Discretización Supervisada No Paramétrica Orientada a la Obtención de Reglas de Decisión"*. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla.

[12] Giradles, R., Aguilar-Ruiz, J., Riquelme, J. *"Aprendizaje Evolutivo de Reglas: Mejoras en la Codificación y Evaluación de Individuos"*. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. LMD27.

[13] César Hervás-Martínez, Cristóbal Romero, Sebastián Ventura. *"Selección de medidas de evaluación de reglas obtenidas mediante programación genética basada en gramática"*. Universidad de Córdoba, Córdoba, España. LMD23.

[14] Carlos Hurtado L. *"Evaluación de Modelos de Clasificación"*. Departamento de Ciencias de la Computación, U de Chile.

[15] Ali Karci and Ahmet Arslan. *"Uniform population in genetic algorithms"*. Istanbul University Engineering Faculty. Journal of electrical & electronics. Elazig/Turkey 2002.

[16] N. Lavrac, P. Flach, B. Zupan. *"Rule Evaluation Measures: A Unifying View"*. ILP-99, LNAI 1634. Springer-Verlag Berlin Heidelberg. 1999.

[17] Liu, B., Hsu, W. & Chen, S. *"Using general impressions to analyze discovered classification rules"*. Proc. 3rd Int. Conf. on Knowledge Discovery & Data Mining (KDD-97), 31-36. AAAI Press, (1997).

[18] Adam Marczyk. *"Algoritmos genéticos y computación evolutiva"*. 2004.

[19] Edgar Noda, Alex A. Freitas, Akebo Yamakami. *"A Distributed-Population Genetic Algorithm for Discovering Interesting Prediction Rules"*. WSC7-2002.

[20] E. Noda, A. Freitas, H.S. Lopes. *"Discovering interesting prediction rules with a genetic algorithm"*. Proc. Congress on Evolutionary Computation, CEC-99. Washington D.C., USA, July 1999. IEEE, Piscataway, NJ. 1999.

[21] P. Berka and I. Bruha. *"Empirical comparison of various discretization procedures"*. Technical Report LISP-95-04, Laboratory of Intelligent Systems, Prage, 1995.

[22] Wesley Romão, Alex A. Freitas y Roberto C. S. Pacheco. *"A Genetic Algorithm for Discovering Interesting Fuzzy Prediction. Rules: applications to science and technology data"*. UEM, DIN-CTC; PUC-PR, PPGIA-CCET; UFSC, PPGEP-CTC. GECCO (2002)

[23] Soltero Domingo, Francisco José; Bodas Sagi, Diego José. *"Clasificadores inductivos para el posicionamiento web"*. El profesional de la información, 2005, enero-febrero, v. 14, n. 1, pp. 4-13.

[24] Weka home page http://www.cs.waikato.ac.nz/~ml/weka/