Learning to Advertise

Anísio Lacerda[1] Marco Cristo[1] Marcos André Gonçalves[1]
Weiguo Fan[2] Nivio Ziviani[1] Berthier Ribeiro-Neto[1 3]

[1]Federal Univ. of Minas Gerais
Dept. of Computer Science
Belo Horizonte, Brazil
{anisio, marco, mgoncalv, nivio, berthier}@dcc.ufmg.br

[2]Virginia Tech
Dept. of Computer Science
Blacksburg VA, USA
wfan@vt.edu

[3]Google Engineering Belo Horizonte
Belo Horizonte, Brazil
berthier@google.com

ABSTRACT

Content-targeted advertising, the task of automatically associating ads to a Web page, constitutes a key Web monetization strategy nowadays. Further, it introduces new challenging technical problems and raises interesting questions. For instance, how to design ranking functions able to satisfy conflicting goals such as selecting advertisements (ads) that are relevant to the users and suitable and profitable to the publishers and advertisers? In this paper we propose a new framework for associating ads with web pages based on Genetic Programming (GP). Our GP method aims at learning functions that select the most appropriate ads, given the contents of a Web page. These ranking functions are designed to optimize overall precision and minimize the number of misplacements. By using a real ad collection and web pages from a newspaper, we obtained a gain over a state-of-the-art baseline method of 61.7% in average precision. Further, by evolving individuals to provide good ranking estimations, GP was able to discover ranking functions that are very effective in placing ads in web pages while avoiding irrelevant ones.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Applications—*Text processing*

General Terms

Algorithms, Experimentation

Keywords

Web Advertising, Genetic Programming

1. INTRODUCTION

The Internet has become one of the most important media for advertising nowadays. It represents the possibility of global exposure to large audiences at very low cost, which attracts great sums in investments in advertising. This situation was different just few years ago, when the failure of many Web companies led to a dropping in supply of cheap venture capital and considerable reduction in on-line advertising investments [29, 30]. According to the Interactive Advertising Bureau (IAB) [18], such reduction caused consecutive declines in quarterly revenues of companies in the US market, beginning with the first quarter of 2001. However, this loss trend has been reversed by the end of 2002. This recovery has coincided with the increasing adoption of a particular Web advertising format, the search advertising. Nowadays, this is the leading format and, by 2010, it will represent a market of more than US$11 billion [23], according to Forrester Research projections.

In search advertising, an advertiser company is given prominent positioning in ad lists in return for a placement fee. Because of this, such methods are called *paid placement strategies*. The most popular paid placement strategy is a non-intrusive technique called *Keyword-targeted advertising* [30]. In this technique, keywords extracted from the user's search query are matched against keywords associated with ads provided by advertisers. A ranking of the ads, which also takes into consideration the amount that each advertiser is willing to pay, is computed. The top ranked ads are displayed in the search result page together with the answers for the user's query.

The success of keyword-targeted advertising has motivated information gatekeepers to offer their ad services in different contexts. For example, relevant ads could be shown to users directly in the pages of information portals. The motivation is to take advantage of the users immediate information interests at browsing time. The problem of matching ads to a Web page that is browsed, which we refer to as *Content-targeted advertising* [21], is different from that of keyword-targeted advertising. In this case, instead of dealing with users' keywords, we have to use the contents of a Web page to decide which ads to display.

A previous work in literature [28] has shown that the use of different pieces of evidence, such as structural information and the contents of the advertiser's page, can impact on the relevance of the ads selected to be displayed. This

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

work, however, did not answer important questions such as how to combine the available pieces of evidence or how much importance should be given to each evidence. This led us to a question: how can we design a ranking strategy for displaying ads according to their relevance by effectively leveraging all the evidence available? Further, given the negative impact of irrelevant ads on credibility and brand of publishers and advertisers, how to design functions that minimize the placement of irrelevant ads, especially when the relevant ones are not available?

To give proper answers for these questions, we propose a new approach to content-targeted advertising based on Genetic Programming (GP). GP is a machine learning technique inspired by biological evolution to find solutions optimized for certain problem characteristics. Our assumption is that GP is able to learn the intrinsic characteristics of the content-targeted advertising problem and use them to provide solutions able to improve the ranking effectiveness.

To validate our GP method we performed experiments using a real ad collection and web pages extracted from a Brazilian newspaper. The results obtained indicate that GP is able to learn ranking functions that are very effective in placing ads in web pages. In particular, our best function provided a gain over state-of-the-art strategies of approximately 61.7% in average precision. Further, GP was able to learn functions that successfully avoid the placement of irrelevant ads by calculating thresholds based on the page where the ads should be placed.

This paper is organized as follows. In Section 2, we provide background information on content-targeted advertising and GP. In Section 3, we describe how we modeled the content-targeted advertising problem using GP. In Section 4, we describe our experiments and report our results. In Section 5, we describe the related work. In Section 6, we present the conclusions.

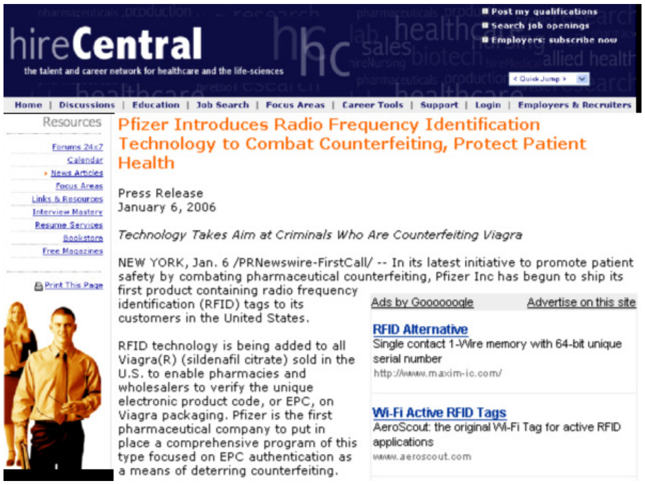
2. BACKGROUND

In this section we present background information on content-target advertising and review the main concepts in Genetic Programming.

2.1 The Content-Targeted Advertising Problem

Content-targeted advertising consists in showing a list of ads in a web page, referred to as the *triggering* page. The ads are expected to be relevant to the users and suitable and profitable to the publishers and advertisers. Therefore, factors that contribute to the order in which the ads are displayed in the lists are: (i) the relatedness and adequacy of ads to the content of the page and (ii) the amount the advertiser is willing to pay for clicks in their ads.

In this work we consider that an ad is composed of three structural parts: a title, a textual description and a hyperlink. In fact, these are the usual components of an ad in search advertising systems. The hyperlink points to a page, called *landing page*, where a transaction can be started. In this page, the user can also find more information related to the ad or to the company, its products and services. Figure 1 illustrates an ad list with two ad slots on the right side of a web page. For the ad in the first ad slot, the title is “RFID Alternative”, the description is “Single contact 1-Wire memory with 64-bit unique serial number”, and the hyperlink points to the site “www.maxim-ic.com”.



Pfizer Introduces Radio Frequency Identification Technology to Combat Counterfeiting, Protect Patient Health

Press Release
January 6, 2006

Technology Takes Aim at Criminals Who Are Counterfeiting Viagra

NEW YORK, Jan. 6 /PRNewswire-FirstCall/ -- In its latest initiative to promote patient safety by combating pharmaceutical counterfeiting, Pfizer Inc has begun to ship its first product containing radio frequency identification (RFID) tags to its customers in the United States.

RFID technology is being added to all Viagra(R) (sildenafil citrate) sold in the U.S. to enable pharmacies and wholesalers to verify the unique electronic product code, or EPC, on Viagra packaging. Pfizer is the first pharmaceutical company to put in place a comprehensive program of this type focused on EPC authentication as a means of deterring counterfeiting.

RFID Alternative
Single contact 1-Wire memory with 64-bit unique serial number
http://www.maxim-ic.com/

Wi-Fi Active RFID Tags
AeroScout: the original Wi-Fi Tag for active RFID applications
www.aeroscout.com

Figure 1: Example of content-based advertising in the page of a company that offers health care jobs. The content of the page is about the usage of an identification technology called RFID. On the right side, we can see ads picked for this page by Google’s content-targeted advertising system.

Besides the visible parts, a set of keywords $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$ is associated with each ad. The keywords comprise one or more words and are used by the advertisers to describe which topics should appear in a web page to display the ads on it. For instance, for the first ad shown in Figure 1, the ad keyword could be “RFID” or “RFID alternative”. To associate a certain keyword k with one of its ads, the advertiser has to bid on k in an auction type system. The more the advertiser bids on k, greater are the chances that its ads will be shown in the ad list of pages in which topic k is present. Notice that the advertisers will only pay for their bids when the users click on their ads. Further, an advertiser can associate several ads with the same product or service. We refer to such group of ads as a *campaign*. Notice that only an ad per campaign should be placed in a web page in order to ensure a fair use of the page advertising space and increase the likelihood that the user will find an interesting ad.

In this work we are particularly interested in the relevance aspect of the content-targeted advertising problem. Given a web collection $\mathcal{D}$ and a set of ads $\mathcal{A}$, our task is to select ads $a_i \in \mathcal{A}$ related to the contents of a Web page $p \in \mathcal{D}$ and rank them according to how relevant they are. The ad list is then built in such way that more relevant ads are placed in top positions and, as far as possible, only one ad per campaign is selected. In the following, we formally define this restriction.

Let $\mathcal{C} = \{C_1, C_2, ..., C_n\}$ be a partition of $\mathcal{A}$ that represents the set of campaigns $C_1, C_2, ..., C_n$. Let $r(a_i, p) \colon \mathcal{A} \times \mathcal{D} \rightarrow \mathbb{R}$ be a function that indicates how relevant is the ad a_i to the triggering page p. Let $\delta_{ijp} \colon \mathbb{N} \times \mathcal{C} \times \mathcal{D} \rightarrow \mathbb{R}$ be a function that represents the relevance score of the i-th top-ranked ad of campaign C_j according to the function r. For instance, if a_s is the second top-ranked ad of campaign C_5, $\delta_{25p} = 0.5$ indicates that $r(a_s, p) = 0.5$. We are interested in finding the function $rank(a_i, p) \colon \mathcal{A} \times \mathcal{D} \rightarrow \mathbb{R}$ that can be used to build rank lists that satisfy the constraint:

$$\forall_{i,j,k|j \neq k} \ (\delta_{ijp} > 0 \wedge \delta_{(i+1)kp} > 0 \Rightarrow \delta_{ijp} > \delta_{(i+1)kp}) \quad (1)$$

As previously mentioned, ad placement systems should minimize the possibility of exhibiting irrelevant ads. Misplacements are particularly common in two situations. First, in spite of the ad and the page being related to the same subject, their mapping is not appropriate. For example, this is the case of placing ads in pages about catastrophes or unethical and illegal advertising. Second, the triggering page is about a topic for which it is hard to find relevant ads. In order to minimize misplacements in such situations, specially the second, a good ranking function should provide reliable relevance estimations such that it would be possible to distinguish the acceptable relevance levels from the not acceptable.

Notice that, in this work, we intend to learn the ranking functions $rank(a_i, p)$, through GP. These ranking functions are designed to optimize overall precision and minimize the number of misplacements.

2.2 Genetic Programming

Genetic programming (GP) [19] is a set of artificial intelligence search algorithms that follows the principles of biological inheritance and evolution. GP is typically used to approximate complex, non-linear functional relationships [19]. Because of the intrinsic parallel search mechanism and powerful global exploration capability in a high-dimensional space, GP has been used to solve a wide range of hard optimization problems that oftentimes have no best known solutions. The overall GP framework for a setting comprising a training and a validation collection is described in Listing 1.

Listing 1: Overall GP Framework.

```
1  Let $\mathcal{T}$ be a training document collection;
2  Let $\mathcal{V}$ be a validation document collection;
3  Let $N_g$ be the number of generations;
4  Let $N_t$ be the number of individuals;
5  $\mathcal{S} \leftarrow \varnothing$;
6  $\mathcal{P} \leftarrow$ Initial random population of individuals;
7  For each generation $g$ of $N_g$ generations do {
8      For each individual $i \in \mathcal{P}$ do
9          $fitness_i \leftarrow fitness(i, \mathcal{T})$;
10     $\mathcal{S}_g \leftarrow$ Get $N_t$ top–ranked individuals of generation
             $g$ according to their fitness;
11     $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_g$;
12     $\mathcal{P} \leftarrow$ New population created by applying genetic
             operators to individuals in $\mathcal{S}_g$;
13 }
14 $\mathcal{F} \leftarrow \varnothing$;
15 For each individual $i \in \mathcal{S}$ do
16     $\mathcal{F} \leftarrow \mathcal{F} \cup \{i, fitness(i, \mathcal{V})\}$;
17 BestIndividual $\leftarrow$ SelectionMethod($\mathcal{F}$, $\mathcal{S}$);
```

In GP, the solution to a problem is represented as an individual (i.e., a chromosome) in a population pool. These individuals are represented by means of complex data structures such as trees, linked lists, or stacks [20]. The length or size of these data structures is not fixed, although it may be constrained by implementation to be within a certain size limit. Initially, the population starts with individuals created randomly as we can see in Listing 1 (line 6). Then they evolve generation by generation through genetic operations (lines 7-13). A fitness function is used to assign the fitness value for each individual (line 9). The fitness value indicates how well they perform in the training examples and it can be used as a means of selecting the best ones (line 10). To evolve the best individuals, genetic operators are applied to them with the aim of creating more diverse and better performing individuals (line 12). Examples of such operators are reproduction, mutation, and crossover. The reproduction operator is used to breed new individuals identical to their parents, the crossover operator takes two individuals (parents) to breed a new one that shares some attributes with each parent, and the mutation operator simulates the deviations that occur in the reproduction process.

The last step in the GP framework presented in Listing 1 consists in determining the best individual to be applied to the test set. The natural choice is the individual with best performance in the training set. However, it might not generalize well due to overfitting[1] during the learning. In order to alleviate this problem the best individuals evolved over N_g generations are applied to a second document collection, which we call a *validation* collection (line 15). Then it is possible to select the individual that presents good performance in both sets, the training and validation (line 17). It is likely to generalize well since it proved to be a good choice in two different document sets.

Therefore, an initial strategy to select the best individual should be to get the one that presents the best average performance in the training and validation sets. However, since the average does not ensure that the selected individual has a balanced performance in the both sets, it would be interesting to consider the standard deviation to correct such a bias.

More formally, we apply the following method to determine our best individual. Let $\bar{f_i}$ be the average performance of individual i in the training and validation collections, and $\sigma(f_i)$ be the corresponding standard deviation. The best individual is given by:

$$\underset{i}{argmax}(\bar{f_i} - \sigma(f_i)) \quad (2)$$

3. MODELING CONTENT-TARGETED ADVERTISING WITH GP

In order to apply GP to the problem of content-targeted advertising we need to define three key components of the GP framework described in Listing 1: the individuals, the genetic operators and the fitness function.

3.1 Individuals

Since we are interested in finding a good ranking function to match ads and pages, as described in Section 2.1, we decided to represent our individual using a tree structure. As observed by [9], a tree based representation allows for easy parsing, implementation, and interpretation. Figure 2 illustrates an individual.

As we can see in Figure 2, the non-leaf nodes in the tree structure ("*", "log", and "/") represent functions applied to the terminals in the leaf nodes. The functions addition

[1] Situation in which the learner may adjust to very specific random features of the training data such that the performance on the training examples still increases while the performance on unseen data becomes worse.

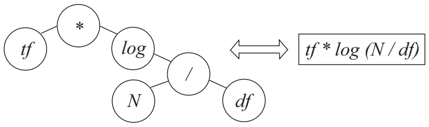


Figure 2: A sample tree representation for a function. Here we show the common TF-IDF weighting scheme.

(+), multiplication (∗), division (/) and logarithm (*log*) are used in our individual representation. They were selected because they provide meaningful operations on relations. For example, matching functions used in Information Retrieval (IR) commonly employ addition and multiplication to reinforce relations in different degrees, division to accommodate inverse relationships, and logarithm to smooth values.

These functions are applied to terminals that are the leaf nodes in the tree structure ("tf", "N", and "df"), as shown in Figure 2. Since in this work we intend, through GP, to discover a single ranking function to find a set of relevant ads with regard to a Web page by combining all or several of the available evidence, the terminals to be used in our representation comprise the information related to this evidence. In other words, the terminals represent statistics about the structural parts of the ads and the information provided by the advertisers such as the keywords associated with the ads and the content of the landing page. Additionally, we use real numbers as terminals to allow fixed weighted factors.

Table 1 describes all the terminals to be used. Notice in this table that P stands for different structural parts of the ads and the information provided by advertisers (keyword, title, description, and landing page), and G indicates whether the ads are grouped. For instance, the feature $tf_{ad,title}$ stands for the number of times a term appears in the title of an ad whereas the feature $tf_{camp,title}$ represents the number of times a term appears in the titles of all the ads of a campaign.

3.2 Genetic Operators

The genetic operators used in our model are those commonly used in GP, that is, mutation, crossover, and reproduction. Notice that, given the representation of our individuals by means of trees, the crossover operator consists in taking two trees and exchanging randomly selected subnodes of these trees forming two new children. Accordingly, the mutation operator was implemented in such a way that a randomly selected subtree is replaced by a new subtree also created randomly.

3.3 Fitness Function

We now define the fitness function that is the objective function GP aims to optimize. The algorithm described in Listing 2 details our fitness evaluation function.

We start by noticing that the ranked lists produced by our random individuals do not satisfy the campaign constraint given by Eq. 1. Thus, in order that function $fitness$ (which corresponds to function *rank* in Section 2.1) can satisfy that constraint, we apply the individual i (which corresponds to function r in Section 2.1) to each campaign in collection $\mathcal{T}$ according to a round robin strategy, as follows. For each campaign, a ranking is built according to the similarity function i (lines 3-4). The top ranked ad of each ranking is then selected till that all the campaigns have been considered (lines 6-9). These ads are sorted according to the relevance scores provided by i and inserted into the final ranked list (lines 10-11). The process is repeated until that no ads remain to be selected (line 5). By doing this, we guarantee that the j-th top-ranked ad of a campaign will always be placed into a page above the $(j+1)$-th top-ranked ad of any other campaign, satisfying the campaign constraint. The fitness value corresponding to individual i is then obtained through the evaluation of the final ranked list (line 12). Note that depending on the evaluation function to be used we can propose different fitness functions. The evaluation functions and corresponding fiteness functions to be used in this work are discussed in the following paragraphs.

Listing 2: Fitness function.

```
1  function fitness(individual $i$, collection $\mathcal{T}$)
2      Let $\mathcal{C} = \{C_1, ..., C_n\}$ be the set of campaigns in $\mathcal{T}$;
3      For all campaigns $C_j \in \mathcal{C}$ do
4          $rlist_j \leftarrow$ Apply $i$ to $C_j$;
5      While exists $j$ such that $|rlist_j| > 0$ do
6          For $j = 1$ to $|\mathcal{C}|$ do
7              If $|rlist_j| > 0$ then
8                  $ad_{top} \leftarrow$ extract top–ranked ad of $rlist_j$;
9                  Insert $ad_{top}$ into $rlist_{temp}$;
10         Sort $rlist_{temp}$;
11         Insert ads in $rlist_{temp}$ into $rlist_{final}$ preserving
               their order;
12     $fvalue \leftarrow$ Evaluate $rlist_{final}$;
13     return $fvalue$;
```

A good ranked list should maximize the placement of relevant ads near to top positions since these are the positions more likely to be clicked by the users [11]. Thus, the evaluation function should take into consideration the number of relevant ads and the order in which they appear, that is, it should be a combination of precision and recall [2], two well-known retrieval measures in IR. An example of such evaluation function is given by:

$$pavg@k = \eta \sum_{i=1}^{k} \left(r(a_i) \times \left(\frac{\sum_{j=1}^{i} r(a_j)}{i} \right) \right) \qquad (3)$$

where $\eta = \frac{1}{k}$ is a normalizing constant used to ensure that $pavg@k$ fits between 0 and 1, k is the number of ads to be displayed in a page, a_i is the i-th top ranked ad, and $r(d) \in \{0, 1\}$ is the relevance score assigned to an ad, being 1 if the document is relevant and 0 otherwise. The relevance information is obtained from users.

This metric is based on the *non-interpolated average precision* (PAVG), a measure commonly used in TREC evaluations [15]. The difference between metrics PAVG and $pavg@k$ is the value of the constant η, which in PAVG is given by the inverse of the total of relevant documents in the collection. By using $\eta = \frac{1}{k}$, we ensure that a ranking function that places relevant ads in all the top positions will receive the maximum $pavg@k$ value equal to 1. In this way, we are able to correctly evaluate functions that suggest a number of ads less than the total of ad slots available. In this work, we refer to the fitness function that uses $pavg@k$ to evaluate its individuals as $f_{pavg@k}$.

Features used	Statistical meaning
$tf_{G,P}$	Number of times the term appeared in the part P of the ad grouped by G.
$tf\_max_{G,P}$	Maximum tf in the part P of the ad grouped by G.
$tf\_avg_{G,P}$	Average tf in the part P of the ad grouped by G.
$tf\_max\_col_{G,P}$	Maximum $tf_{G,P}$ in the entire collection.
$length_{G,P}$	Number of terms in the part P of the ad grouped by G.
$n_{G,P}$	Number of distinct terms in the part P of the ad grouped by G.
$df_{ad,P}$	Number of ads in the collection the term appeared in the part P.
$df\_max_{ad,P}$	Maximum $df_{ad,P}$.
$df_{camp,P}$	Number of campaigns in the collection the term appeared in the part P.
$df\_max_{camp,P}$	Maximum $df_{camp,P}$.
N_{ad}	Number of ads in the collection.
N_{camp}	Number of campaigns in the collection.
N	Real constant randomly generated by GP.

Table 1: Terminals used in the GP framework for content-targeted advertising

Other goal that we want to accomplish with our fitness functions is to reward ranking functions that minimize the placement of irrelevant ads. As mentioned before, these ads should be avoided since they contribute to a negative perception by the users on the credibility and brand of publishers and advertisers. A possible solution to this problem is to consider the ranking values provided by the GP individuals as estimations of how relevant the ads are to the triggering page. By doing so, we can set threshold values to distinguish acceptable relevance levels from non-acceptable ones.

Thus, our problem is now finding a matching function that provides reliable estimations in a spectrum in which a threshold value can be set to separate relevant ads from non relevant ones. Our assumption is that GP is able to find such functions. Thus, given a certain threshold level t, we modify our evaluation function such that it rewards individuals that tends to place relevant ads above t and nonrelevant ads below t. Accordingly, it punishes individuals that tends to place irrelevant ads above t and relevant ads below t. Our second evaluation metric is given by:

$$pavg@k_t = \frac{1 + k_1\ r_{at} + k_2\ n_{bt}}{1 + k_3\ r_{bt} + k_4\ n_{at}}\ pavg@k, \qquad (4)$$

where k_1, k_3, k_2, and k_4 are the weights associated with the number of relevant ads above (r_{at}) and below (r_{bt}) the threshold and non relevant ads below (n_{bt}) and above (n_{at}) the threshold, respectively.

Notice that in our experiments we give more weight to n_{at} since we want specially to avoid the placement of irrelevant ads in the top positions. In particular, we use $k_1 = k_3 = k_2 = 1$ and $k_4 = 2$.

An important remaining issue is how to define the threshold value t. In this work we define $t = v_{min} + k_t\ (v_{max} - v_{min})$, where v_{min} and v_{max} are the minimum and maximum values given by the ranking function. The constant k_t is the relative position in the spectrum the GP individual should consider a point of low confidence. In our experiments we use $k_t = 0.3$. In other words, our new fitness functions will reward ranking functions in which the minimum score assigned to a relevant ad corresponds to 30% of $(v_{max} - v_{min})$.

Notice that, in fact, it is not possible to know the values of v_{min} and v_{max} because we deal with randomly generated functions. As a consequence we define these limits by inspecting the rank values provided by our random individuals. In this study we adopt two different strategies to estimate the limit values. In the first, we use the maximum value given to a certain page as v_{max} and the minimum value as v_{min}. Thus, we have different thresholds for different pages. We refer to the fitness function that uses $pavg@k_t$ to evaluate its individuals and calculate thresholds for each page as f_{local}. A possible disadvantage of f_{local} is that it tends to suggests, at least, one ad per page. In the second strategy we use the maximum value given to an individual as v_{max} and the minimum value as v_{min}. In this case we have only one threshold value for a function. In this work we refer to the fitness function that uses $pavg@k_t$ to evaluate its individuals and calculate thresholds for each individual as f_{global}. Contrary to f_{local}, f_{global} is more likely to suggest no ads to a certain page.

4. EXPERIMENTS

In this section we describe the experiments and present the results.

4.1 Sampling and Data Sets

To evaluate our ad placement framework, we used a test collection built from a set of 100 pages extracted from a Brazilian newspaper. These are our triggering pages. They were crawled in such a way that only the contents of their articles were preserved. As we have no preference for particular topics, these pages cover subjects as diverse as culture, local news, international news, economy, sports, politics, agriculture, cars, children, real estate, computers and internet, TV, travels, and economy.

To obtain a set of relevant ads for our test collections, we adopted the same pooling method used to evaluate the TREC Web-based collection [16]. In other words, for each of our 100 triggering pages, we selected the top three ranked ads provided by each of the ten ad placement methods proposed in [28]. These ads were obtained from a real case ad collection composed of 93,972 ads grouped in 2,029 campaigns provided by 1,744 advertisers. With these ads, advertisers associated a total of 68,238 keywords[2]. In this collection, only one keyword is associated with each ad. This makes campaigns very important since they are used by the advertisers to associate several keywords with a product or service. As a result of the pooling method, a total of 1,860 distinct ads were selected. They were then inserted into pools corresponding to each triggering pages. Each pool contained an average of 15.81 ads. All the ads were submitted to a manual evaluation by a group of 15 subjects. Each

[2] Data in the portuguese language provided by an on-line ad company that operates in Brazil.

subject was asked to evaluate the ads selected to each page according to its relevance to the pages. The average number of relevant ads per page pool was 5.15.

Since our experiment can be qualified as a supervised learning task, we follow the three data-sets design [7, 24]. In other words, the 2,337 evaluated pairs of ads and documents resulting of the pooling process were used to built training, test, and validation sets. For this, we randomly split the data into three parts. We used 50 pages (and its corresponding ads) for training, 30 pages for validation, and 20 pages for test. As previously mentioned, the introduction of the validation dataset is to help alleviate the problem of overfitting of GP on the training data and select the best generalizable individual. All the results reported in this work are based on the test data set.

4.2 Setup

We learned on the training sample using different parameters. We noticed that a small population size and different rates for the genetic operations produce better results. The size of the populations used in our experiment was fixed at 750 individuals. The maximum depth of the tree used to represent an individual was set as 17. In all experiments related here, the populations were created using four different random seeds and were allowed to evolve for 30 generations. This number was determined empirically. The random seeds used were 245, 37383, 322443, and 6758. As in [19], we used crossover, mutation, and reproduction rates of 85%, 10%, and 5%, respectively. We tested our GP framework using the three fitness functions described in Section 3.3. Experiments for each function were run four times using the different random seeds. The best result among the four runs is reported and used for comparison.

4.3 Evaluation and Baseline

We present the results of our experiments considering that a triggering page offers three ad slots. We report figures using *pavg*@3 (Eq. 3, with $k = 3$), for the case in which the methods assign exactly three ads per page. For the cases in which they are allowed to assign less than three ads, we use *pavg*@k (Eq. 3) and *pavg*@k_t (Eq. 4). In all the cases, as in [28], we also report the number of hits and ads suggested per ad slot. We call *hit* the placement of a relevant ad.

We compare the results of our GP ranking functions with those obtained by the AAK_H method described in [28]. This method consists in using a cosine similarity function to match the triggering page to the ad. Besides its title and description, the content of the ad, as used by AAK_H, includes the content of the keyword and the landing page associated with it. Further, this method requires that all the terms in the ad keyword be present in the triggering page to the ad to be considered a good matching. Amongst the methods presented in [28], which take into account only the ad title, description, keywords, and landing page, AAK_H is the best. Given these pieces of evidence, note that, as far as we know, this is the best method found in the literature. This makes AAK_H an ideal baseline since our GP individuals make use of the same body of evidence.

4.4 Results

In this section we present the results of experiments with exactly three ads per page and with possibly less than three ads per page.

Methods	Hits/Suggestions				*pavg*@3	
	#1	#2	#3	Total	Score	Gain
AAK_H	9/20	5/20	9/20	23/60	0.314	–
GP1	14/20	11/20	7/20	32/60	0.508	+61.7%

Table 2: Performance comparison between the best individual evolved from the optimization of $f_{pavg@k}$ (GP1) and baseline method (AAK_H). Columns labelled #1, #2, and #3 indicate the total of hits and suggestions for the first, second, and third ad slots, respectively.

4.4.1 Experiments with exactly three ads per page

As we can see in Table 2, our best GP individual (GP1), reached a performance of 50.8% in *pavg*@3. This corresponds to a gain of 61.7% when compared with our baseline. An interesting characteristic of GP1 is its successful performance in the first ad slot which is the one more likely to be clicked by the users [11].

Figure 3 displays the evolution along 30 generations of the population from which GP1 was selected. For each generation we can see the ten best individuals sorted according to the performance of their fitness function ($f_{pavg@k}$). The figure shows a remarkable difference in the performance of the individuals when we compare training and test sets. This is due to overfitting. The individuals applied to the training set tend to learn very specific characteristics not found in the test set. As a consequence, the best individuals of the training set are not so good in the test set. However, by selecting the best individual using Eq. 2, we were able to get a ranking function that generalizes well.

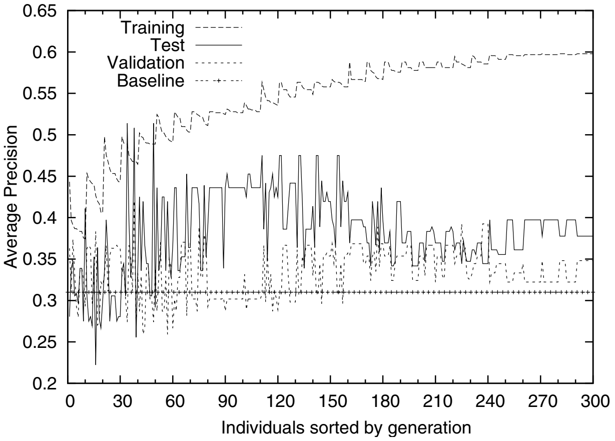


Figure 3: Evolution Process for 300 individuals in 30 generations. Notice that each ten individuals correspond to one generation.

4.4.2 Experiments with possibly less than three ads per page

Table 3 compares the performance of the best individuals obtained by GP that have evolved to avoid placing irrelevant ads according to threshold values. In this table, GP2 is the individual evolved from the optimization of f_{local}. The line corresponding to this individual shows performance figures for the case in which the threshold value is not taken into consideration. That is, all the top ads selected by GP2 are evaluated independently of their ranking scores. The line started with GP2+thr corresponds to the same individual for the opposite case, that is, the threshold value is taken into consideration. Similarly, GP3 evolved from the optimization of f_{global} and its corresponding performance figures are shown for the cases where the threshold was used (GP3+thr) and was not used (GP3).

Notice in Table 3 that GP2 and GP3 present better performance than the baseline with gains of 37.2% and 9.6%, respectively, for the *pavg*@k metric. These results, however, are worse than those obtained with our best individual, GP1. This is due in part to the fact that more precise individuals tend to misplace ads less frequently and, consequently, they have less opportunities to be rewarded by correctly placing irrelevant ads below a certain threshold.

When we analyze the performance of the individuals after applying the thresholds, we notice an improvement for GP2+thr and no difference for GP3+thr. For instance, method GP2+thr was able to avoid placing twelve irrelevant ads in the third slot with the loss of only five ads. When considering the metric *pavg*@k_t, the gain of GP2+thr over GP2 was approximately of 16%. This allows us to conclude that GP was able to learn functions that avoid the placement of irrelevant ads and present good overall performance for the case in which different thresholds are obtained for each page. Conversely, for the case in which a unique global threshold has to be used, GP was not able to learn good ranking functions.

5. RELATED WORK

The success of search advertising has motivated research in many topics related to targeted advertising. Examples of these studies include the comparison of ranking strategies [11], the characterization of fake traffic in order to detect frauds [5], the proposal of tools for keyword suggestion [3], and the design and implementation of a large-scale targeted advertising system [1].

In particular, the relevance aspect of the ranking strategies has attracted attention. This is not surprising since many works in advertising research have emphasized the importance of relevant associations for consumers [26] and how irrelevant ads can turn off users and relevant ads are more likely to be clicked on [11]. As a result, some works have tried to determine how to take advantage of the available evidence to improve the relevance of the selected ads. For instance, studies on keyword matching showed that the nature and size of the keywords have impact on the likelihood of an ad to be clicked [25]. Relevance is also the focus of the authors in [28] which proposed several strategies for ranking ads in content-targeted advertising. These strategies took into consideration the contents of structural parts of the ad and additional information obtained from web pages other than the triggering page. Examples of these pages are the landing pages or web pages obtained by means of a probabilistic model. They showed that considering the contents of the ad structural parts and external pages can improve the relevance of the selected ads. In contrast to that work, we propose to *learn* the best ranking strategies in order to effectively leverage all the evidence available while minimizing the placement of irrelevant ads. For this, we use GP.

GP has been applied to several IR topics in recent years, such as query induction, representation, and optimization [4, 17, 22], document clustering and classification [14, 31], and document ranking [13, 27]. From these, many works [6–8, 10] have applied GP to discover ranking functions. For example, success has been reported in applying GP to find ranking functions optimized to specific queries in the information routing task [7]. Similarly, GP has also been successfully used in the ad-hoc retrieval task [10]. In fact, this work is inspired on this prior research in ranking function discovery. But it differs significantly in several important aspects. Since we intend to find ranking functions for content-targeted advertising, we deal with specific characteristics of this problem not found in classical IR tasks previously studied. For instance, content-targeted advertising presents different kinds of evidence, the possibility of taking advantage of campaign clustering statistics, and specific ranking related issues such as campaign placement restrictions and impact of irrelevant ads.

6. CONCLUSIONS

In this paper we proposed and tested a new framework for associating ads with web pages based on GP. In particular, given the importance of relevance for content-target advertising systems, our GP method aimed to learn functions able to select the more relevant ads given the available evidence. By using a real ad collection and web pages from a Brazilian newspaper, we obtained a gain over our baseline method of 61.7%. Further, by evolving individuals to provide good ranking estimations, GP was able to discover ranking functions that are very effective in placing ads in web pages while avoiding the irrelevant ones.

In the future we intend to provide more extensive and comprehensive analysis of our models and expand them in order to contemplate additional evidence and consider other important aspects of the content-targeted advertising problem. Regarding model analysis, we intend to study how different threshold tuning strategies impact on the learning and effectiveness of our GP framework. We also plan to perform more extensive comparison of our method with other machine learning techniques, such as the SVM-based approach [8]. Future plans also include a detailed study of why GP ranking functions outperform other techniques in this task. Regarding new models, we intend to evolve functions that take into consideration the category information associated with ads and pages. More important, we plan to expand our models to yield ranking functions that combine the relevance and monetary aspects of the problem by considering the amount the advertiser is willing to pay for the placement of their ads.

7. ACKNOWLEDGMENTS

This work was supported in part by the GERINDO Project grant MCT/CNPq/CT-INFO 552.087/02-5, by the CNPq grant 30.5237/02-0 (Nivio Ziviani), by the 5S-QV project grant MCT/CNPq/CT-INFO 551013/2005-2, by the CNPq grant 300188/95-1 (Berthier Ribeiro-Neto), by grant 10023-UFMG/RTR/PRPQ/RECEM-DOUTORES/04 (sub-grant 32-PROGRAMACAO GENETICA). Marco Cristo is supported by Fucapi, Manaus, AM, Brazil.

Methods	Hits/Suggestions				$pavg@k$		$pavg@k_t$	
	#1	#2	#3	Total	Score	Gain(%)	Score	Gain(%)
AAK_H	9/20	5/20	9/20	23/60	**0.31**	–	–	–
GP2	10/20	11/20	8/20	29/60	0.43	+38.7	**1.12**	–
GP2+thr	10/20	10/18	3/ 3	23/41	0.49	+58.1	1.30	+16.1
GP3	10/20	9/20	5/20	24/60	0.34	+9.6	**0.59**	–
GP3+thr	10/20	9/20	5/19	24/59	0.34	+9.6	0.59	0.0

Table 3: Performance of the best individuals evolved from the optimization of f_{local} (GP2) and f_{global} (GP3). Columns labelled #1, #2, and #3 indicate total of hits and suggested ads for the first, second, and third ad slots, respectively. Note that the values in gain columns are relative to boldface values in the corresponding left columns.

8. REFERENCES

[1] G. Attardi, A. Esuli, and M. Simi. Best bets: thousands of queries in search of a client. In *Proceedings of the 13th international WWW conference on Alternate track papers & posters*, pages 422–423, New York, NY, USA, 2004. ACM Press.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, 1st edition, 1999.

[3] J. J. Carrasco, D. Fain, K. Lang, and L. Zhukov. Clustering of bipartite advertiser-keyword graph. In *Workshop on Clustering Large Datasets, 3th IEEE International Conference on Data Mining*, Melbourne, Florida, USA, November 2003. IEEE Computer Society Press. Available at `http://research.yahoo.com/publications.xml`.

[4] O. Cordon, F. Moya, and C. Zarco. A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 6(5):308–319, Aug. 2002.

[5] E. Eneva. Detecting invalid clicks in online paid search listings: a problem description for the use of unlabeled data. In T. Fawcett and N. Mishra, editors, *Workshop on the Continuum from Labeled to Unlabeled Data, 20th International Conference on Machine Learning*, Washington DC, USA, August 2003. AAAI Press.

[6] W. Fan, E. A. Fox, P. Pathak, and H. Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search. *JASIST*, 55(7):628–636, 2004.

[7] W. Fan, M. D. Gordon, and P. Pathak. Discovery of context-specific ranking functions for effective information retrieval using genetic programming. *TKDE-04*, 16(4):523–527, 2004.

[8] W. Fan, M. D. Gordon, and P. Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *IPM-04*, 40(4):587–602, 2004.

[9] W. Fan, M. D. Gordon, and P. Pathak. Genetic programming-based discovery of ranking functions for effective web search. *Journal of Management Information Systems*, 21(4):37–56, Spring 2005.

[10] W. Fan, M. D. Gordon, P. Pathak, W. Xi, and E. A. Fox. Ranking function optimization for effective web search by genetic programming: An empirical study. In *Proc. of HICSS-04*, pages 105–112, Hawaii, 2004.

[11] J. Feng, H. Bhargava, and D. Pennock. Implementing paid placement in Web search engines: computational evaluation of alternative mechanisms. *INFORMS Journal on Computing*, 2006. To be published.

[12] D. Gleich and L. Zhukov. SVD based term suggestion and ranking system. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 391–394, Brighton, UK, November 2004. IEEE Computer Society.

[13] M. Gordon. Probabilistic and genetic algorithms for document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988.

[14] M. D. Gordon. User-based document clustering by redescribing subject descriptions with a genetic algorithm. *JASIS*, 42(5):311–322, 1991.

[15] D. K. Harman. Overview of the fourth text retrieval conference TREC-4. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 1–24, Gaithersburg, Maryland, USA, November 1996. NIST Special Publication 500-236.

[16] D. Hawking, N. Craswell, and P. B. Thistlewaite. Overview of TREC-7 very large collection track. In *The Seventh Text REtrieval Conference (TREC-7)*, pages 91–104, Gaithersburg, Maryland, USA, November 1998.

[17] J.-T. Horng and C.-C. Yeh. Applying genetic algorithms to query optimization in document retrieval. *Inf. Process. Manage.*, 36(5):737–759, 2000.

[18] IAB and PricewaterhouseCoopers. IAB internet advertising revenue report, April 2005. Available at `http://www.iab.net/2004adrevenues`.

[19] J. R. Koza. *Genetic programming: On the programming of computers by natural selection.* MIT Press, Cambridge, 1992.

[20] W. B. Langdon. *Data Structures and Genetic Programming: Genetic Programming + Data Structures = Automatic Programming!* Kluwer, Boston, 1998.

[21] K. Lee. The SEM content conundrum. ClickZ Experts, July 2003. Available at `http://www.clickz.com/experts/search/strat/article.php/2233821`.

[22] C. Lopez-Pujalte, V. P. Guerrero-Bote, and F. de Moya-Anegon. Order-based fitness functions for genetic algorithms applied to relevance feedback. *J. Am. Soc. Inf. Sci. Technol.*, 54(2):152–160, 2003.

[23] K. Maddox. Forrester reports advertising shift to online, May 2005. Available at `http://www.btobonline.com/article.cms?articleId=24191`.

[24] T. M. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1996.

[25] OneUpWeb. How keyword length affects conversion rates, January 2005. Available at `http://www.oneupweb.com/landing/keywordstudy_landing.htm`.

[26] J. Parsons, K. Gallagher, and K. D. Foster. Messages in the medium: An experimental investigation of Web Advertising effectiveness and attitudes toward Web content. In J. Ralph H. Sprague, editor, *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6050, Washington, DC, USA, 2000. IEEE Computer Society.

[27] P. Pathak, M. Gordon, and W. Fan. Effective information retrieval using genetic algorithms based matching function adaptation. In *Proceedings of the 33rd Hawaii International Conference on System Science*, Hawaii, USA, 2000.

[28] B. Ribeiro-neto, M. Cristo, E. S. de Moura, and P. B. Golgher. Impedance coupling in content-target advertising. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 496–500, Salvador, Bahia, Brazil, July 2005.

[29] M. Weideman. Ethical issues on content distribution to digital consumers via paid placement as opposed to website visibility in search engine results. In *The 17th ETHICOMP*, pages 904–915. Troubador Publishing Ltd, April 2004.

[30] M. Weideman and T. Haig-Smith. An investigation into search engines as a form of targeted advert delivery. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 258–258. South African Institute for Computer Scientists and Information Technologists, 2002.

[31] B. Zhang, Y. Chen, W. Fan, E. A. Fox, M. Gonalves, M. Cristo, and P. Calado. Intelligent gp fusion from multiple sources for text classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 477–484, New York, NY, USA, 2005. ACM Press.