Optimal Design of Web Information Contents for E-Commerce Applications

Alfredo Milani[1], Valentino Santucci[1], Clement Leung[2]

[1]University of Perugia, via Vanvitelli, 1 – Perugia, Italy
{milani@unipg.it, valentino.santucci@dipmat.unipg.it}

[2]Hong Kong Baptist University, Hong Kong
{clement@comp.hkbu.edu.hk}

Abstract. Optimization of web content presentation poses a key challenge for e-commerce applications. Whether considering web pages, advertising banners or any other content presentation media on the web, the choice of the appropriate structure and appearance with respect to the given audience can obtain a more effective and successful impact on users, such as gathering more readers to web sites or customers to online shops. Here, the collective optimization of web content presentation based on the online discrete Particle Swarm Optimization (PSO) model is presented. The idea behind online PSO is to evaluate the collective user feedback as the PSO objective function which drives particles' velocities in the hybrid continuous-discrete space of web content features. The PSO coordinates the process of sampling collective user behaviour in order to optimize a given user-based metric. Experiments in the online banner optimization scenario show that the method converges faster than other methods and avoid some common drawbacks such as local optima and hybrid discrete/continuous features management. The proposed online optimization method is sufficiently general and may be applied to other web marketing or business intelligence contexts.

Keywords: Web marketing optimization, collective behaviour mining, collaborative intelligence.

1 Introduction and Related Work

In Web-based information systems adopting an appropriate structure and appearance with respect to the given user community can raise its impact and effectiveness [11][12]. This is particularly true for e-commerce applications and social networking sites whose aims are to attract more customers or increase the rate of participation. Online advertising is a key enabler of e-commerce and in an advertising banner scenarios, a designer has to compose a web banner by considering a variety of options such as different background colours, available pictures of the product, presentation phrases, font types, and sizes. Typically, the banner designer employs his own skills and his model of the target customer in order to design what he consider the best eye catching or attention catching advertising. The only way for the designer to know if the banner is effective is to submit it to the web and consider users reactions. Managing

E. Gelenbe et al. (eds.), *Computer and Information Sciences*, Lecture Notes in Electrical Engineering 62, DOI 10.1007/978-90-481-9794-1_64,

this interactive process for a large number of users is nearly impossible for a banner designer, on the other hand an automatic optimization [13] of the content presentation can exploit the feedback of a large number of online users. Some applications [1][13] has been proposed which try to select the optimal presentation using a voting mechanism (i.e. user feedback, such as number of clicks) among a fixed set of candidate ones, e.g. a set of candidate banners, or by tuning some features parameters by randomly generating candidates to vote [2]. The limit in the first case is that the optimal solution could not be in the fixed set of candidates, while a purely random strategy can hardly find an optimal solution because of combinatorial explosion, especially in presence of features with nearly continuous values, such as colours or image sizes. Particle swarm optimization (PSO) [3][4] technique has recently emerged as an effective strategy for a variety of multidimensional optimization problems. PSO uses the algorithmic metaphor of the dynamic of swarm behaviour in order to coordinate a set of particles, i.e. computational units, which move through a given features domain space. Evolutionary approaches to adaptive web content selection has been previously used in the field of web newspapers and multimedia information retrieval in [5][6] [7].

In this paper, we formulate a strategy based on PSO for generating candidate presentation instances, which eventually converge to the optimal content presentation. A single particle has a current position in the features space, i.e. it represents a content presentation instance. Each particle in the swarm sends a candidate presentation to a set of online users which provide a feedback. Particle positions are updated according to PSO strategy based on the user feedback. Experiments show that the PSO strategy for web content presentation is effective and converges very fast, minimizing the number of sampled candidate presentations, i.e. minimizing the number of non-optimal presentations delivered.

2 PSO for Web Marketing

The principle underlying PSO has been introduced in [3]. The metaphor receives its inspiration from particles models of objects and simulation of collective behaviour of flocks of birds. In PSO, a swarm is composed of a set of particles $P=\{p_1,p_2,\ldots,p_k\}$. The position of a particle in the n-dimensional space $\Re^n$ corresponds to a candidate solution of a given optimization problem represented by an objective function $f: \Theta \rightarrow \Re$, with $\Theta \subseteq \Re^n$, to be maximized (or minimized).

At any step t, each particle p_i has associated a position $x_{i,t}$, a velocity $v_{i,t}$, where position and velocity are n-dimensional vectors, and $b_{i,t}$ the particle personal best, i.e. the best position of p_i has ever visited until time step t. Moreover, particles are interconnected in a network and can communicate only with their neighbors l_i; in this way each particle can maintains the best position ever found among his l_i's neighbors denoted by $l_{i,t}$.

Each particle in the swarm moves according to its velocity. Position is updated by the vector expression $x_{i,t+1} = x_{i,t} + v_{i,t+1}$, while velocity is updated by $v_{i,t+1} = \omega v_{i,t} + \varphi_1 \beta_{1,t}(b_{i,t} - x_{i,t}) + \varphi_2 \beta_{2,t}(l_{i,t} - x_{i,t})$, where the weights respectively represent the inertia ω, the acceleration factors φ_1, φ_2 and the random factors $\beta_{1,t}$, $\beta_{2,t}$ which are distributed in

[0,1]. The contribution ($b_{i,t}$ - $x_{i,t}$), the distance from the personal best, has been interpreted as a cognitive component, while ($l_{i,t}$ - $x_{i,t}$) is a social component.

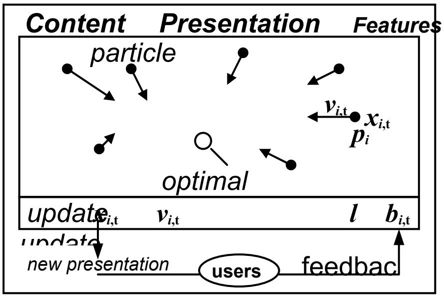
A number of variations to PSO has been proposed for velocity updating or other aspects. A very common one assumes that particles are connected by a complete network and in this case $l_{i,t}$ are substituted by a global l which can be maintained more efficiently. This simple variation is the one used in our approach.

As pointed out in [10] and [3] PSO seems to benefit from the local monotony of objective function f in continuous search spaces, but the same property does not hold in the discrete feature spaces generated by combinatorial problems. Solutions have been proposed for discrete PSO since [8] and more recently [9], while a distinction should be made between *ordered discrete* features, for which a discrete approximation of local monotony hold, and *pure combinatorial* feature for which it does not. Let consider the *discrete* feature *temperature* T, whose domain D_T , is the finite set of ordered values D_T=*{veryCold, cold, cool, mild, warm, veryWarm, hot, veryHot}*, if the best value so far for f has been found in position T=*veryWarm* then is likely that f does not differ much for values close to *veryWarm*. On the other hand a *pure combinatorial* feature domain has a finite number of values which cannot be ordered according to some notion of distance for which an approximation of continuity properties of objective function f hold. For instance the feature *product name* D_{PN}={*WeatherPlus, allWeatherInfo, EveryWeather, FastForecasting, Sun&Rain*}, containing candidate names for a weather forecasting web site, is *pure combinatorial* since it contains values which have no significant ordering relationship inducing local monotony in objective function f. Moreover the distinction among continuous, discrete ordered and pure combinatorial domains is not so sharp: discrete ordered domains of very large size can be easily managed by continuous approximation, while discrete ordered domain of small cardinality are better regarded as pure combinatorial.

3 An Online PSO Model for Web Marketing Content Presentation

The search space here consists of admissible content presentations. The content presentation search space is described by a feature vector $C=[c_1,\dots,c_n]$ with $c_i \in D_i$, where D_i are possible alternatives provided by the content presentation designer. In content presentation problems, the domains D_i are, in general, a mix of continuous, discrete and pure combinatorial domains. The particle swarm algorithm proposed here uses a fully connected particle swarm. At each iteration each particle generates a new candidate presentation configuration by moving to a new position $x_{i,t}$ in the search space D. The evaluation of the objective function $f(x_{i,t})$ is realized by submitting the candidate presentation to web users and measuring their feedback.

The users feedback is used in order to determine the personal best, absolute best and in order to perform velocity update. The algorithm aims at maximizing the feedback function. A scheme of the online algorithm is shown in Fig. 1. The set P of particles is initially distributed in a random way in the search space. If the content designer has his own preferred or best candidate it is directly assigned to one particle. Personal best and global best are initially assigned to zero for all particles (i.e. no feedback observed). Velocity updating has an important role in the proposed algorithm. Since it is supposed to have a hybrid continuous/discrete features space, different update functions are used for different classes of dimensional domains. The purpose is to exploit the local continuity for continuous and discrete domains and emphasize the exploration for pure combinatorial domains. In the following is described the update phase of velocity and position which varies, as mentioned above, depending on the domain to which the feature belongs.



Features(#×size)	C_{eucl}	P_{schw}	C_{schw}	P_{ack}	C_{ack}
Mix (6×2^6)	860	1.00	980	0.62	1360
Mix (6×2^6)	1000	0.76	1680	1.00	5800
Cont.(8×2^7)	800	0.76	1820	0.40	2480
Discrete 1(6×2^7)	220	1.00	320	1.00	9640
Discrete 2(8×2^7)	360	1.00	51820	0.80	18580
Discrete 3(10×2^7)	520	0.00	---	0.00	---
Pure Comb.1(3×2^6)	2600	1.00	220	1.00	14060
Pure Comb.2(4×2^6)	44800	1.00	18000	0.57	108920
Pure Comb.3(5×2^6)	---	0.85	122560	0.00	---

Fig. 1. (a) Online PSO scheme for content presentation and (b) Experimental results

Continuous Features

Position and velocity of continuous features are updated according to the updating functions given above. Features with many discrete values, like *pictureSize*, are considered discrete approximation of continuous position and velocity. Out of bound exceptions are managed by randomly restarting the particle dimension.

$x_{i,t}$

Pure Combinatorial Features

The domain of a *pure combinatorial* feature consists of a discrete set of values for which an ordering is not defined. In this case a random extraction which combines both domain exploration and the behaviour of classical PSO approach is used.

For each particle p and for each pure combinatorial feature domain D_c, an appropriate *probability distribution* $P_{Dc,p}$ over the values of D_c is built. Let d_b , d_p, d_l the values of feature D_c respectively for the *global best* position, the *current* particle position and the particle *local best* position, then the distribution is defined as

$$P_{Dc,p}(d_p) = (1+\omega)/n \cdot 1/N_{Dc} \qquad P_{Dc,p}(d_b) = (1+\varphi_1)/n \cdot 1/N_{Dc}$$
$$P_{Dc,p}(d_l) = (1+\varphi_2)/n \cdot 1/N_{Dc} \qquad P_{Dc,p}(d) = 1/n \cdot 1/N_{Dc} \quad \forall\, d \in D_c \;\; d \notin \{ d_b , d_p, d_l \}$$

where $n = |D_c|$ and $N_{Dc,p}$ is a normalization factor: $N_{Dc} = 1+(\omega+\varphi_1+\varphi_2)/n$. The probability distribution used to extract the next particle value can be interpreted as considering all the values equiprobable, except d_b , d_p, d_l, whereas the amplification factors ω, φ_1 and φ_2 gives a greater probability to d_b , d_p and d_l . In other words the amplification factors respectively express, the tendency to remain in the current position (inertial factor ω), and the tendency to move toward the global best (social factor φ_1) or the local best positions (cognitive factor φ_2).

Discrete Ordered Features

Discrete ordered features are content features for which a total order exists. Let $D_d=\{d_1 ,\ldots, d_n\}$ the values of a discrete ordered domain D_j then a probability distribution $P_{Dd,p}$ is built similarly to the case of pure combinatorial features, but reflecting the additional property that "close feature values have close probabilities". The probability distribution is obtained by smoothing the probabilities in the contour of the "centers" d_p d_b and d_l . Initially a quantity $1/n$ is assigned to each value d_i and then the values in the centers and in the points to their left and to their right are incrementally amplified. Let $\alpha \in \{1+\omega, 1+\varphi_1, 1+\varphi_2\}$ the amplification factor for a center d_k and let $\beta = \alpha/(\lambda+1)$ where is 2λ is the amount of values centered in d_k whose probability will be amplified with *smoothness* β. All the λ values d_{k-j} (d_{k+j}) at the left (right) of index k will be amplified with parameter $\alpha_j = \beta * |\lambda + 1 - j|$, process is iterated for every center d_p d_b and d_l and the final quantities are then normalized to obtain the probability distribution $P_{Dd,p}$. The parameter λ determines the interval of features values where the $(1+\alpha_j)$ amplification is applied, it is easy to see that values near the *global optimum*, *local optimum*, and *current* particle values tend to be preferred, while the initial distribution $1/n$ ensures that each domain value has a non null probability of being selected.

4 Experiments and Discussion

Experiments have been carried out using the hidden values technique developed in [1] where the PSO algorithm make external call to different online user feedback functions. Three different *user feedback functions* has been experimented: (1) the *Euclidean feedback* f_{eucl}, which returns the Euclidean distance from the optimal solution, (2) f_{schw} a relaxed version of the *Schwefel*'s double sum function [2] and (3) the *Ackley* function f_{ack} [2].

$$f_{eucl}(x) = \sqrt{\sum_{i=1}^{D} x_i^2} \tag{1}$$

$$f_{schw}(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2 \tag{2}$$

$$f_{ack}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i^2\right)\right) + 20 + e \tag{3}$$

[1]. Feedback function f_{eucl} and f_{schw} are unimodal functions, while f_{ack} is a multimodal function, in order to investigate the behaviour of online PSO approach both with single optimal value and multiple local optima. For each of the three benchmark *user feedback* functions, nine different types of domain combinations with homogeneous and hybrid features have been tested: the hybrid features mix (*Mixed Banner Real*), an hybrid artificial case (*Mixed Banner Artificial)* and six cases of homogeneous features of different type and size which has been used for performance comparison (*Continuous, Discrete1-3, Purely Combinatorial 1-3)*.

The results shows that the performance of convergence probability drops with purely combinatorial domains, or with an increasing number of discrete ordered features (f_{schw} and f_{ack} do not converge with 10 discrete ordered features of size 128) with respect to continuous domains. On the other hand the algorithm always determines the optimal content design in all hybrid cases, *Mixed Banner Real* and *Mixed Banner Artificial*, it also converge (62%) with "difficult" user models like the *Ackley* function f_{ack} where multiple local minima are present.

The convergence speed for the hybrid cases is also quite encouraging: the $\underline{C_{best}}$ value of NFE in all the user feedback models lies within a range of one to six thousands fitness evaluations, i.e. user contacts, which appears to be a realistic size for many online applications.

5 Conclusions

A method based on online PSO has been introduced and experimented in order to model the process of optimizing the design of content presentations by evaluating the collective feedback of online users, which is regarded as PSO fitness.

The PSO particles moves in the features space of the presentation objects, each new position correspond to a new content presentation, the PSO dynamics guided by user feedback guarantees convergence toward the optimal content presentation.

A novel method to manage *purely combinatorial* and *discrete ordered* feature domains has been introduces. The method is based on building probabilities distributions biased toward the global optimal, the local optimal and the current particle feature values. The biased probabilities reflect the typical PSO inertial, local and global search strategies. Experiments shows that the online PSO method converges for a realistic hybrid feature mix after a realistic number of user feedback evaluation.

References

[1] M. Bolin, M. Webber, P. Rha, T. Wilson, R.C. Miller. "Automation and customization of rendered web pages". In Proc. 18th ACM Symp. on User interface Software and Technology, ACM Press, New York, USA 2005, pp. 163-172.

[2] K. Marriott, B. Meyer, L. Tardif. "Fast and efficient client-side adaptivity for SVG". In Proc.11th Int.Conf.onWorld Wide Web, Hawaii, USA, May 07, 2002.

[3] J. Kennedy, R. Eberhart. "Particle swarm optimization". In Proc. of IEEE Conf. on Neural Networks, IEEE Press, 1995, pp. 1942-1948.

[4] R. Poli, J. Kennedy, T. Blackwell. "Particle swarm optimization. An overview.". Swarm Intelligence, 1(1): 33-57, 2007.

[5] A. Milani. "Online genetic algorithms", in International Journal of Information Theories and Applications, n.1 Vol.11, pp.20-28, (2004), ISSN 1310-0513.

[6] J.H. Lin, C.K. Wang, C.H. Lee (Taiwan). "Particle swarm optimization for web newspaper layout problem" in J.T. Yao (Ed) Web Technologies, Applications, and Services – 2006 pp. 524-026 ISBN0-88986-575-2.

[7] C. Leung, A. Chan. "Community adaptive search engines" in Int.J.of Advanced Intelligence Paradigms(IJAIP), Special Issue on Intelligent Techniques for Personalization and Recommendation, Inderscience, 2008, ISSN 1755-0386.

[8] J. Kennedy, R. Eberhart. "A discrete binary version of the particle swarm algorithm". In Proc. of the IEEE Conf. on Systems, Man, Cybernetics, IEEE Press, 1997, pp. 4104-4108.

[9] X.H. Zhi, X.L. Xing, Q.X. Wang, Zhang. "A discrete PSO method for generalized TSP problem", 2004. Proceedings of 2004 Int. Conference on Machine Learning and Cybernetics Vol. 4, 26-29 Aug. 2004, pp. 2378-2383.

[10] R. Poli. "Analysis of the publications on the applications of particle swarm optimisation", J. of Artificial Evolution and Applications, Article ID 685175, 10 pages, 2008. ISSN 1755-0386.

[11] W.S. Chan, A. Milani, C.H.C. Leung, J. Liu. "An architectural paradigm for collaborative semantic indexing of multimedia data objects", LNCS in Springer Verlag.

[12] A. Bunt, G. Carenini, C . Conati. "Adaptive co/ntent presentation for the web", in P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The Adaptive Web. Berlin: Springer, (2007), pp. 409-432.

[13] A. Jameson. "Adaptive interfaces and agents", in Jacko, J., Sears, A. (eds.): Human-Computer Interaction Handbook, Erlbaum, Mahwah, NJ (2003), pp. 305-330.