Int. J. Communications, Network and System Sciences, 2014, 7, 497-507
Published Online December 2014 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2014.712050

Playing against Hedge

Miltiades E. Anagnostou[1], Maria A. Lambrou[2]

[1]School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
[2]Department of Shipping, Trade and Transport, University of the Aegean, Chios, Greece
Email: miltos@central.ntua.gr, mlambrou@aegean.gr

Received 10 October 2014; revised 20 November 2014; accepted 1 December 2014

Abstract

Hedge has been proposed as an adaptive scheme, which guides the player's hand in a multi-armed bandit full information game. Applications of this game exist in network path selection, load distribution, and network interdiction. We perform a worst case analysis of the Hedge algorithm by using an adversary, who will consistently select penalties so as to maximize the player's loss, assuming that the adversary's penalty budget is limited. We further explore the performance of binary penalties, and we prove that the optimum binary strategy for the adversary is to make greedy decisions.

Keywords

Hedge Algorithm, Adversary, Online Algorithm, Greedy Algorithm, Periodic Performance, Binary Penalties, Path Selection, Network Interdiction

1. Introduction

The problems of adaptive network path selection and load distribution have often been considered as games that are played simultaneously and independently by agents controlling flows in a network. A possible abstraction of these and other related problems is the bandit game. In the *multi-armed bandit* game [1] a player chooses one out of N strategies (or "machines" or "options" or "arms"). A loss or penalty (or a reward, which can be modeled as a negative loss) ℓ_i is assigned to each strategy i $\left(i = 1, 2, \cdots, N\right)$ after each round of the game.

An agent facing repeated selections will possibly try to exploit the so far accumulated experience. A popular algorithm that can guide the agent in each selection round is the *multiplicative updates* algorithm or *Hedge*. In this paper we calculate the worst possible performance of Hedge by using the adversarial technique, *i.e.* we investigate the behavior of an intelligent adversary, who tries to maximize the player's cumulative loss. In Section 1 we describe Hedge; in Section 2 we give a rigorous formulation of the adversary's problem; in Section 3 we give a recursive solution; and in Section 4 we present sample numerical results. Finally, in Section 5 we

How to cite this paper: Anagnostou, M.E. and Lambrou, M.A. (2014) Playing against Hedge. *Int. J. Communications, Network and System Sciences*, **7**, 497-507. http://dx.doi.org/10.4236/ijcns.2014.712050

explore binary adversarial strategies. Our main result is that the greedy adversarial strategy is optimal among binary strategies.

1.1. The Bandit Game

In a generalized bandit game the player is allowed to play mixed strategies, *i.e.* to assign a fraction p_i (such that $\sum_{i=1}^{N} p_i = 1$) of the total bet to option i, thereby getting a loss equal to $L = \sum_{i=1}^{N} p_i \times \ell_i$. Alternatively, p_i can be interpreted as a probability that the player assigns the bet on option i. In the "bandit" version only the total loss L is announced to the player, while in the "full information" version the penalty vector $(\ell_1, \ell_2, \cdots, \ell_N)$ is announced.

A game consists of T rounds; a superscript t marks the tth $(t = 0, \cdots, T-1)$ round. Apparently the player will try to minimize the total cumulative loss

$$\sum_{t=0}^{T-1} L^t = \sum_{t=0}^{T-1} \sum_{i=1}^{N} p_i^t \times \ell_i^t \tag{1}$$

by controlling the bet distribution, *i.e.* by properly selecting the variables p_i^t. We use the additional assumption that the loss budget is limited in each round by setting the constraint $\sum_{i=1}^{N} \ell_i^t = 1$. Clearly a player's goal is to minimize his or her total cumulative loss. An extremely lucky player, or a player with "inside information", would select the minimum penalty option in each round and would put all his or her bet on this option, thereby achieving a total loss equal to $\sum_{t=0}^{T-1} \min_i \ell_i^t$.

1.2. The Hedge Algorithm

Quite a few algorithmic solutions, which will guide the player's hand in the full information game, have appeared in the literature. Freund and Schapire have proposed the *Hedge* algorithm [2] for the full information game. Auer, Cesa-Bianchi, Freund and Schapire have proposed the *Exp*3 algorithm in [3]. Allenberg-Neeman and Neeman proposed a Hedge variant, the *GL* (Gain-Loss) algorithm, for the full information game with gains and losses [4]. Dani, Hayes, and Kakade have proposed the *GeometricHedge* algorithm in [5], and a modification was proposed by Bartlett, Dani *et al.* in [6]. Recently Cesa-Bianchi and Lugosi have proposed the ComBand algorithm for the bandit version [7]. A comparison can be found in [8].

Hedge maintains a vector $w^t = (w_1^t, w_2^t, \cdots, w_n^t)$ of weights, such that $w_i^t \geq 0$ ($t = 0, 1, \cdots, T-1$, and $i = 1, 2, \cdots, N$). In each round t Hedge chooses the bet allocation according to the normalized weight $p_i^t = w_i^t \big/ \sum_{i=1}^{N} w_i^t$. When the opponent reveals the loss vector of this round, the next round weight w^{t+1} is determined so as to reflect the loss results, *i.e.* $w_i^{t+1} = w_i^t \beta^{\ell_i^t}$ for some fixed β, such that $0 \leq \beta \leq 1$.

In [9] Auer, Cesa-Bianchi, Freund and Schapire have proved that the expected Hedge performance and the expected performance of the best arm differ at most by $O\left(\sqrt{TN \ln N}\right)$. Freund and Schapire [2] have given a loss upper bound, which relates the total cumulative loss with the total loss of the best arm.

1.3. Competitive Analysis

The competitive analysis of an algorithm $\mathcal{A}$, which in this paper is Hedge, involves a comparison of $\mathcal{A}$'s performance with the performance of the optimal offline algorithm. In the bandit game the optimal offline algorithm, *i.e.* the optimal player's decisions given the sequence of all penalties in advance, is trivial. In a given round the player can just bet everything on the option with the lowest penalty.

According to S. Irani and A. Karlin (in Section 13.3.1 of [10]) a technique in finding bounds is to use an "adversary" who plays against $\mathcal{A}$ and concocts an input, which forces $\mathcal{A}$ to incur a high cost. Using an adversary is just an illustrative way of saying that we try to find the worst possible performance of an online algorithm. In our analysis the adversary tries to maximize Hedge's total loss by controling the penalty vector (under a limited budget).

1.4. Interpretations and Applications

In this section we offer some interpretations from the areas of 1) communication networks and 2) transportation. The general setting of course involves a number of options or arms, which must be selected by a player without any knowledge of the future.

Bandit models have been used in quite diverse decision making situations. In [11] He, Chen, Wand and Liu have used a bandit model for the maximization of the revenue of a search engine provider, who charges for advertisements on a per-click basis. They have subsequently defined the "armed bandit problem with shared information"; arms are partitioned in groups and loss information is shared only among players using arms of the same group. In [12] Park and Lee have used a multi-armed bandit model for lane selection in automated highways and autonomous vehicles traffic control.

1.4.1. Traffic Load Distribution

This first application example can take multiple interpretations, which always involve a selection in a competitive environment, in which competition is limited. It can be seen as 1) a path selection problem in networking, 2) a transport means (mode) choice or path selection problem, 3) a computational load distribution problem, which we mention in the end of this section. Firstly, we describe the problem in the context of networking.

Consider N similar independent paths (in the simplest case just N parallel links), which join a pair of nodes $\mathcal{A}$, $\mathcal{B}$. A traffic volume equal to Q is sent from $\mathcal{A}$ to $\mathcal{B}$ in consecutive time periods or rounds by a population of agents. Q is the same in each round, but the allocation of Q to paths, *i.e.* $\left(Q_1^t, Q_2^t, \cdots, Q_N^t\right)$ such that $\sum_{i=1}^{N} Q_i = Q$, is different in each round t. An agent A produces a constant amount of traffic equal to A, such that $q \ll Q$, in T consecutive rounds, and allocates a part equal to q_i $\left(\sum_{i=1}^{t} q_i = q\right)$ to the i th path in round t. The average delay (or cost) experienced by A's traffic in the t th round is proportional to $\sum_{i=1}^{N} Q_i^t q_i^t$, if we assume a linear delay (or cost) model. Linear models are used for simplicity in network analysis [13] and can be realistic if a network resource still operates in the linear region of the delay vs. load curve, e.g. when delay is calculated in a link, which operates not very close to capacity. Agent A aims at minimizing the total delay for its own traffic and may use Hedge to determine the quantities q_i^t in round t, assuming that A knows the performance of its own traffic in each path in the past time period. Note that the maximum delay in a round occurs if A puts the whole q in a single path together with the whole traffic of the competition, *i.e.* with Q; then A's average delay in this round equals Q. On the contrary, if Q is evenly distributed in all paths, A's allocation decision does not really matter, as the average will be equal to $\sum_i (q_i/q) \times (Q/N) = Q/N$. Of course the minimum delay in a round will occur if A puts the whole q in an empty path, thereby achieving a zero delay.

The above problem can also be formulated as a more general problem of distributing workload over a collection of parallel resources (e.g. distributing jobs to parallel processors). A. Blum and C. Burch have used the following motivating scenario in [14]: A process runs on some machine in an environment with N machines in total. The process may move to a different machine at the end of a time interval. The load ℓ_i^t, which will be found on a machine i at time round t is the penalty felt by the process.

1.4.2. Interdiction

Although an adversary is usually a "technical" (fictional) concept, which serves the worst case analysis of online algorithms, in some environments a real adversary, who intentionally tries to oppose a player, does exist. An example is the interdiction problem.

We present a version of the interdiction problem in a network security context. An attacker attacks N resources (e.g. launches a distributed denial of service attack on nodes, servers, etc., see [15]) by sending streams of harmful packets to resource i at a rate w_i (where $i = 1, \cdots, N$ and $\sum_i w_i$ is constant). A defender assigns a defense mechanism of intensity ℓ_i (e.g. a filter that is able to detect and avoid harmful packets with a probability proportional to ℓ_i) to resource i. At the end of a time interval T, e.g. one day, both the attacker and the defender revise the flows and the distribution of defense mechanisms to resources respectively, based on past performance.

Similar interpretations exist in transportation network environments, as in border and custom control, including illegal immigration control. An interdiction problem formulation can be used in a maritime transport security context: pirates attack the vessels traversing a maritime route. In [16] Vanek *et al*. assign the role of the player to the pirate. The pirate operates in rounds, starting and finishing in his home port. In each round he selects a sea area (arm) to sail to and search for possible victim vessels. A patrol force distributes the available escort resources to sea areas (arms), and pirate gains are inversely proportional to the strength of the defender's forces on this area. Naval forces reallocate their own resources to sea areas.

2. Problem Formulation

In this paper we aim at finding the worst case performance of Hedge. Effectively, we try to solve the following problem:

Problem 1. *Given a number of options* N, *an initial normalized weight vector* $\boldsymbol{w}=(w_1,w_2,\cdots,w_N)$, *and a Hedge parameter* β, *find the sequence* $\boldsymbol{\ell}^0$, $\boldsymbol{\ell}^1$, $\cdots$, $\boldsymbol{\ell}^{T-1}$ *that maximizes the player's total cumulative loss*

$$L_{H(\beta)}=\sum_{t=0}^{T-1}\sum_{i=1}^{N}p_i^t\ell_i^t \tag{2}$$

where $\boldsymbol{\ell}^t=(\ell_1^t,\cdots,\ell_N^t)$ is the penalty vector in round t $(t=0,1,\cdots,T-1)$, such that $\sum_{i=1}^{N}\ell_i^t=1$, and the t th round penalty weights p_i^t are updated according to

$$w_i^t=w_i^{t-1}\beta^{\ell_i^{t-1}}=w_i\beta^{\sum_{\tau=0}^{t-1}\ell_i^\tau},\quad p_i^t=\frac{w_i^t}{\sum_{i=1}^{N}w_i^t}\quad(t\geq 1) \tag{3}$$

for $t=1,\cdots,T-1$ and $p_i^0=w_i$. □

Clearly the objective function (2) is a function of a) the N initial weights w_i, and b) the $N\times T$ variables ℓ_i^t, and c) β. Due to the normalization of both weights and penalties there are $(N-1)\times(T+1)+1$ independent variables in total. In the following we use $L^{T-1}\left(w_1,\cdots,w_N;\ell_1^0,\cdots,\ell_N^0,\cdots,\ell_1^{T-1},\cdots,\ell_N^{T-1}\right)$ or $L^{T-1}\left(\boldsymbol{w};\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1}\right)$ instead of $L_{H(\beta)}$ whenever it is necessary to refer to these variables.

3. Recursion

Assuming that a given round starts with weights $\boldsymbol{w}=(w_1,\cdots,w_N)$ and the adversary generates penalties $\boldsymbol{\ell}=(\ell_1,\cdots,\ell_N)$, the next round will will start with weights $\boldsymbol{W}(\boldsymbol{w},\boldsymbol{\ell})=(W_1(\boldsymbol{w},\boldsymbol{\ell}),\cdots,W_N(\boldsymbol{w},\boldsymbol{\ell}))$ where

$$W_i(\boldsymbol{w},\boldsymbol{\ell})=\frac{w_i\beta^{\ell_i}}{\sum_{j=1}^{N}w_j\beta^{\ell_j}}\quad(i=1,2,\cdots,N). \tag{4}$$

Then, the total loss of a T round game, which starts with weights $\boldsymbol{w}$, can be written as the sum of the losses of a single round game, which starts with weights $\boldsymbol{w}$, and a $T-1$ round game, which starts with weights $\boldsymbol{W}(\boldsymbol{w},\boldsymbol{\ell})=(W_1(\boldsymbol{w},\boldsymbol{\ell}),\cdots,W_N(\boldsymbol{w},\boldsymbol{\ell}))$, as follows:

$$L^{T-1}\left(\boldsymbol{w};\boldsymbol{\ell}^0,\boldsymbol{\ell}^1,\cdots,\boldsymbol{\ell}^{T-1}\right)=L^0\left(\boldsymbol{w};\boldsymbol{\ell}^0\right)+L^{T-2}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\ell}^0\right);\boldsymbol{\ell}^1,\cdots,\boldsymbol{\ell}^{T-1}\right). \tag{5}$$

Note that the term L^{T-2}, which expresses the contribution of the last T rounds, depends only on the updated weights provided by the initial round. Such a Markovian property can be generalized in the following sense: A T_1+T_2 round game can be seen as consisting of a T_1 round game g_1 followed by a T_2 round game g_2, whose initial weights are the final weights of g_1, and no more details about g_1 are passed to g_2. Assuming that the solution to Problem 1 is $L_{\max}^{T-1}(\boldsymbol{w})=\max_{\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1}}L^{T-1}\left(\boldsymbol{w};\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1}\right)$ the following recursive formula for $L_{\max}^{T-1}(\boldsymbol{w})$ can be derived from (5):

$$L_{\max}^{T-1}(\boldsymbol{w}) = \max_{\boldsymbol{\ell}}\left[L^0(\boldsymbol{w};\boldsymbol{\ell}) + L_{\max}^{T-2}(\boldsymbol{W}(\boldsymbol{w};\boldsymbol{\ell}))\right] \tag{6}$$

where $\boldsymbol{\ell}^0 = \boldsymbol{\ell}$ is the penalty vector chosen by the adversary in the initial round.

The optimal penalties can be computed also recursively. Let $\lambda^{T-1;t}(\boldsymbol{w}) = \left(\lambda_1^{T-1;t}(\boldsymbol{w}),\cdots,\lambda_N^{T-1;t}(\boldsymbol{w})\right)$, where $\lambda_i^{T-1;t}(\boldsymbol{w})$ denotes the i th optimal penalty of the i th option in the t th round of a T round game (starting with weights $\boldsymbol{w}$). The optimal penalty of the initial round $(t=0)$ is apparently equal to the value of $\boldsymbol{\ell}$, which optimizes (6). Therefore

$$\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w}) = \arg\max_{\boldsymbol{\ell}}\left[L^0(\boldsymbol{w};\boldsymbol{\ell}) + L_{\max}^{T-2}(\boldsymbol{W}(\boldsymbol{w};\boldsymbol{\ell}))\right]. \tag{7}$$

In all other rounds $t = 1,2,\cdots,T-1$ the optimal penalties are such that the total loss of the rest of the game is maximized, *i.e.* such that $L_{\max}^{T-2}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ is achieved. Since the total loss $L_{\max}^{T-2}(\boldsymbol{w})$ is achieved by using penalties $\boldsymbol{\lambda}^{T-2;t}(\boldsymbol{w})$, the total loss $L_{\max}^{T-2}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ is realized by using $\boldsymbol{\lambda}^{T-2;t}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right)$ instead. Therefore

$$\boldsymbol{\lambda}^{T-1;t+1}(\boldsymbol{w}) = \boldsymbol{\lambda}^{T-2;t}\left(\boldsymbol{W}\left(\boldsymbol{w},\boldsymbol{\lambda}^{T-1;0}(\boldsymbol{w})\right)\right) \quad (t = 0,1,\cdots,T-2). \tag{8}$$

4. Two Option Games and Numerical Results

This section we exploit the recursive methodology, which has been presented in the previous section, in order to provide some numerical results for two option games. We compare these results with available bounds in the literature. We consider $N = 2$, *i.e.* two option games. We keep only the independent penalties ℓ_1^t in the extended notation and use the more compact version $L^{T-1}\left(w_1;\ell_1^0,\ell_1^1,\cdots,\ell_1^{T-1}\right)$. As an example, the loss of a single round game is given by

$$L^0(w;\ell) = w\ell + (1-w)(1-\ell). \tag{9}$$

Also, since the initial weights are $\boldsymbol{w} = (w,1-w)$, we simplify the maximum cumulative loss $L_{\max}^{T-1}(\boldsymbol{w})$ to $L_{\max}^{T-1}(w)$. Assuming losses $\ell_1^0 = \ell$ and $\ell_2^0 = 1-\ell$, the next round will will start with weights $W(w,\ell)$ and $1-W(w,\ell)$, where

$$W(w,\ell) = \frac{w\beta^\ell}{w\beta^\ell + (1-w)\beta^{1-\ell}}. \tag{10}$$

Then (6) is simplified to

$$L_{\max}^{T-1}(w) = \max_{\ell}\left[L^0(w;\ell) + L_{\max}^{T-2}(W(w,\ell))\right] \tag{11}$$

where $\ell^0 = \ell$ is the penalty chosen by the adversary for the first option in the initial round.

The iteration starts from $L_{\max}^0(w)$, *i.e.* the loss of a single round game. In such game the adversary controls a single penalty variable, as the loss is given by (9). Apparently the adversary will choose binary values, *i.e.* $\ell = \ell_1^0 = 1$ $\left(\ell_1^0 = 0\right)$ if $w = w_1 > 1/2$ $(w_1 < 1/2)$, and the maximum total loss is $L_{\max}^0(w) = \max\{w,1-w\}$, *i.e.*

$$L_{\max}^0(w) = \begin{cases} 1-w, & \text{if } 0 \le w \le \dfrac{1}{2}, \\ w, & \text{if } \dfrac{1}{2} \le w \le 1. \end{cases} \tag{12}$$

The graph of $L_{\max}^0(w)$ appears as the lowest V-shaped "curve" in **Figure 1**. The fact that the $L_{\max}^0(w)$ is a piecewise linear function of w with a breakpoint (*i.e.* a sudden change in its slope), creates even more breakpoints in $L_{\max}^1(w)$, $L_{\max}^2(w)$ and so on. Therefore, while it is possible to use the aforementioned recursion in

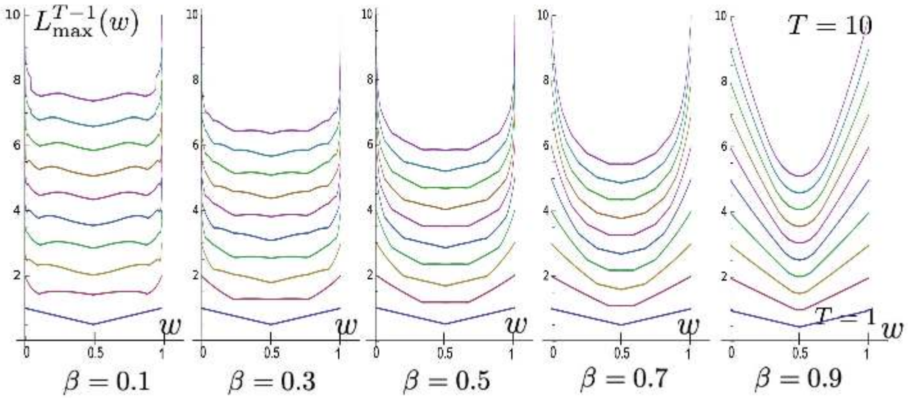


Figure 1. Plot of $L_{\max}^{T-1}(w)$ (maximum loss in a T round game) vs. w for $\beta = 0.1, 0.3, \cdots, 0.9$ and $T = 1, 2, \cdots, 10$.

order to find analytical expressions for the maximum total loss and the associated penalties, the analysis becomes quite complicated even for small values of the number of rounds T (*i.e.* in a $T+1$ round game). We omit this tedious analysis and present numerical results based on the recursive methodology given above.

Instead we have implemented a numerical computation based on (11). $L_{\max}^{T-1}(w)$ is approximated by $K+1$ samples in the interval $[0,1]$, *i.e.* by $L_{\max}^{T-1}(i\Delta w)$, where $i = 0, 1, \cdots, K$ and $\Delta w = 1/K$. In the same way the functions $L^0(w;\ell)$ and $W(w,\ell)$ are represented by $(K+1)^2$ samples $L^0(m\Delta w; n\Delta\ell)$ and $W(m\Delta w, n\Delta\ell)$, where $\Delta w = \Delta\ell$. We have used $K = 1000$. Initially we create $L_{\max}^0(i\Delta w)$ $(i = 0, 1, \cdots, K)$ by using (9). We use the result as input to (11) and create $(L_{\max}^1(i\Delta w))$. Then we use the already calculated L^0 and L^1 in (11) to calculate L^2, then L^0 and L^2 to calculate L^3, and so on. In **Figure 1** we show $L_{\max}^{T-1}(w)$ as a function of the initial weight $w_1 = w$ in games with up to ten rounds $(T = 1, \cdots, 10)$ for different values of β. Observe that the shape of $L_{\max}^{T-1}(w)$ is more "interesting" for "unreasonably" small values of β.

The optimal penalties can be determined by using formulas (7) and (8) for $N = 2$. In **Figure 2** we draw one of the curves of **Figure 1** together with the respective optimal penalties. The final round optimal penalty (*i.e.* $\lambda^{3;3}(w)$ in this example) is certain to be binary, since the adversary will assign $\ell_i^3 = 1$ to the option i with the greatest weight factor. However, the penalties $\lambda^{3;0}(w)$ and $\lambda^{3;1}(w)$ of the first two games are clearly non-binary.

5. Binary and Greedy Schemes

The penalty values in the first two rounds in the example of **Figure 2** prove that the adversary's optimal penalties are not necessarily binary. However, in this example β is "unnaturally" close to 0, as in practical Hedge implementations β is chosen close to 1; this choice achieves a more gradual adaptation to losses. Both experimental and analytical evidence show that the optimal penalties tend rapidly to binary values as β approaches 1. Effectively, it seems that results very close to optimum can be achieved by a "binary adversary", *i.e.* an adversary that will resort to binary values only.

On the other hand the optimal adversarial policy with binary penalties can be found exhaustively as

$$L_{\text{maxbin}}^{T-1}(\boldsymbol{w}) = \max_{(\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1})\in S^T} L^{T-1}(\boldsymbol{w};\boldsymbol{\ell}^0,\cdots,\boldsymbol{\ell}^{T-1})$$

where S is a set of N binary vectors $(b_1, b_2, \cdots, b_N)$ such that $\sum_{i=1}^{N} b_i = 1$, *i.e.* only one component equals 1. Apparently, the complexity of this calculation grows with N^T. However, in the following we show that the optimal binary adversary is in fact the "greedy adversary", The latter achieves binary optimality in linear time.

A "greedy adversary" is eager to punish the maximum weight option as much as possible in each round. Thus

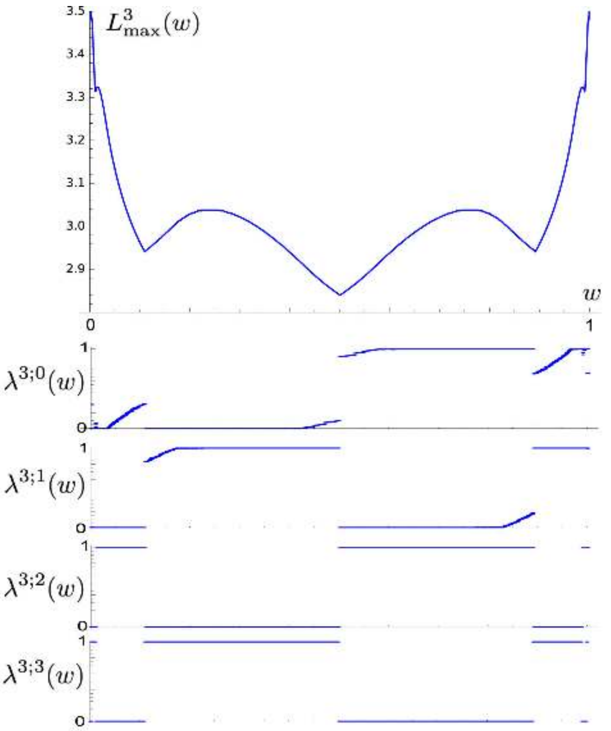


Figure 2. Plot of $L^3_{\max}(w)$ (maximum total loss of a 4 round game) vs. w for $\beta = 0.1$, together with the optimal penalties $\lambda^{3;t}$ $(t = 0,1,2,3)$.

the adversary will assign exactly one unit of penalty to the maximum current weight option, and zero penalties to all other options. Given a sufficient number of rounds (say t_0), it easy to see that the weights of an N option game are "equalized" so that any two weights p_i^t, p_j^t are such that $p_i^t / p_j^t < \beta$ for $t \geq t_0$. When equalization is achieved, a periodic phenomenon starts and the greedy penalties form a rotation scheme.

5.1. Greedy Behavior

We explore the greedy pattern in a two option game that can easily be generalized to N options. Assuming initial weights w_1, w_2 $(w_2 = 1 - w_1)$ such that $w_1 > 1/2 > w_2$, a greedy adversary will choose

$\ell_1^0 = \ell_1^1 = \cdots = \ell_1^{t_0 - 1} = 1$, $\ell_1^{t_0} = 0$ iff $w_1 \beta^{t_0 - 1} > w_2 > w_1 \beta^{t_0}$, where $t_0 \geq 1$ (having assumed $w_1 > w_2$). At t_0 the weight of the second option becomes for the first time greater than the weight of the first option, and a loss equal to 1 is assigned to the second option. Therefore, in the next step $t_0 + 1$ the weights (before normalization) are $w_1 \beta^{t_0}$ and $w_2 \beta$, or equivalently $w_1 \beta^{t_0 - 1}$ and w_2 for the second time. In the next round they become $w_1 \beta^{t_0}$ and w_2 again, and in general they oscillate between these two pairs periodically. Therefore the total loss for $t \geq t_0$ in a pair of subsequent rounds is equal to

$$L_p = \frac{w_1 \beta^{t_0}}{w_1 \beta^{t_0} + w_2 \beta} + \frac{w_2}{w_1 \beta^{t_0} + w_2}. \tag{13}$$

The value of t_0 is determined by the initially assumed inequality, and since t_0 ought to be integer $t_0 = \lceil (\ln w_2 - \ln w_1) / \ln \beta \rceil$. The loss in the first t_0 steps $(t = 0,1,\cdots,t_0 - 1)$ is equal to

$$w_1 + \sum_{\tau=1}^{t_0 - 1} \frac{w_1 \beta^\tau}{w_1 \beta^\tau + w_2}.$$

Therefore, for an even positive integer $T-t_0$ the total loss in T steps is

$$L_{H(\beta)} = w_1 + \sum_{\tau=1}^{t_0-1} \frac{w_1\beta^{\tau}}{w_1\beta^{\tau}+w_2} + \frac{T-t_0}{2}\left[\frac{w_1\beta^{t_0}}{w_1\beta^{t_0}+w_2\beta} + \frac{w_2}{w_1\beta^{t_0}+w_2}\right].$$

In a game with more than two options it is straightforward to show that in the "steady" (periodic) state weights tend to become equal, *i.e.* almost equal to $1/N$, where N is the number of options. Consequently, the loss is given by $L_{H(\beta)} \approx T/N$ in a T round game.

5.2. Optimality of the Greedy Behavior

The following proposition provides a simple polynomial solution to the problem of finding the optimal binary adversary.

Proposition 1. *The greedy strategy is optimal for the adversary among all strategies with binary penalties.* □

Proof: Due to normalization of weights and penalties, in the proof we mention only option 1 weights and penalties. Assuming an initial weight ω and penalties $\ell_1^0, \ell_1^1, \cdots, \ell_1^{n-1}$ in the first n rounds, the weight, which emerges before the (n + 1)th round is $\omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$, where $L = \sum_{i=0}^{n-1}\ell_1^i$. Effectively, two options are available to the adversary in each step, either i) to assign a penalty equal to 1, which will produce an incremental loss equal to $\omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$, and will update the weight to $\omega\beta^{L+1} / \left(\omega\beta^{L+1} + 1 - \omega\right)$ or ii) to assign a zero penalty, which will produce a loss equal to $1 - \omega\beta^L / \left(\omega\beta^L + 1 - \omega\right)$ and an updated weight equal to $\omega\beta^{L-1} / \left(\omega\beta^{L-1} + 1 - \omega\right)$. Define $f(x) \equiv \omega\beta^x / \left(\omega\beta^x + 1 - \omega\right)$.

This looks like a new game, in which the adversary is the player. The player's status is determined by a real number x, and possible rewards are $f(x)$ and $1-f(x)$. If the player opts for $f(x)$, this will bring him to a new status $x+\delta$. If he opts for $1-f(x)$, this will bring him to $x-\delta$. In our case $\delta = 1$. Note also that $f(-\infty) = 1$, $f(+\infty) = 0$, and $f(0) = \omega$. Moreover, if ξ_0 is the root of $f(x) = 1/2$ (or $f(x) = 1 - f(x)$), then $f(x) \geq 1/2$ for $x \leq \xi_0$, and $f(x) \leq 1/2$ for $x \geq \xi_0$. It is easy to prove that there is an odd symmetry around $(\xi_0, 1/2)$, *i.e.* $f(\xi_0 + x) + f(\xi_0 - x) = 2f(\xi_0) = 1$, and $f(x)$ is concave in (∞, ξ_0), while it is convex in (ξ_0, ∞).

Assume that $\omega \geq 1/2$, then $f(0) = \omega \geq 1/2$, and $\xi_0 \geq 0$. If the current status of the player is x_1, and $x_1 < \xi_0$, the greedy behavior is to move $\left\lceil (x_1 - \xi_0)/\delta \right\rceil$ times to the right, which (unless T is too short) will bring the player to a point x_2 such that $x_2 \geq \xi_0$. If $x_2 > \xi_0$, then $1 - f(x_2) > \frac{1}{2} > f(x_2)$ and the greedy player must choose $1 - f(x_2)$ and move back to $x_2 - \delta < \xi_0$. Effectively, this starts an oscillation between $x_2 - \delta$ and x_2, which will last until the end of the game. In the following we prove that this behavior is optimal, in spite of the fact that profits around ξ_0 are low.

The main idea behind this sketch of proof is that a retreat (with consequent low profits $1 - f(x)$ is never a good investment for the future. Assume x_1 as the player's status, and T steps (rounds) remain until the end of the game, while $x_1 + T\delta < \xi_0$. The player executes M forward steps, *i.e.* $x_i = x_1 + i\delta$, $i = 0, 1, \cdots, M-1$, with rewards $f(x_i)$. Then, $M-1$ backward steps with gains $1 - f(x_i)$ are executed; consequently x_1 is reached again. In the rest of the game, *i.e.* until the T th step, greedy selections are made. This course of events is shown on curve (a) in **Figure 3**, where the dots mark the rewards achieved (and some dots have been vertically displaced by a small amount so as to be distinguishable from other dots at the same position). If greedy selections had been made all the way, the course of events would be as shown by curve (b).

If y_i describes the status of the adversary on the greedy curve (b) at the i th step and x_i the status on curve (a), then $f(x_i) = f(y_i)$ for $i = 0, \cdots, M-1$. Furthermore, $f(x_{3M+i}) = f(y_{M+i})$. Therefore the difference between the cumulative reward on curve (b) and curve (a) is

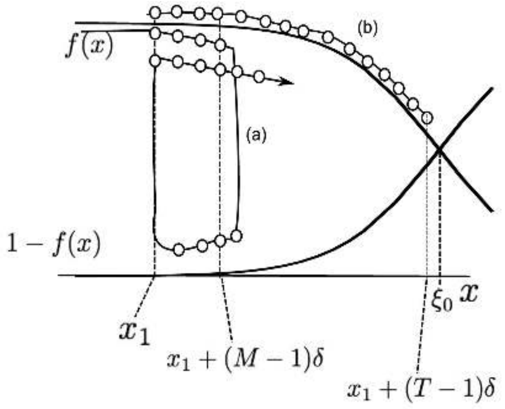


Figure 3. Sample paths of player behavior, which are used in the proof of Proposition 1.

$$\Delta R=\sum_{i=1}^{T}\left[f\left(y_i\right)-f\left(x_i\right)\right]=\sum_{i=T-2M+1}^{T}f\left(x_1+i\delta\right)-\left(\sum_{i=0}^{M-1}f\left(x_1+i\delta\right)+\sum_{i=1}^{M}\left[1-f\left(x_1+i\delta\right)\right]\right)$$
$$=\sum_{i=T-2M+1}^{T}f\left(x_1+i\delta\right)-\left[M+f\left(x_1\right)-f\left(x_1+M\delta\right)\right].$$

Effectively we need to show that $\Delta R\geq 0$. First, let us make some observations and explore other variations of $\Delta R\geq 0$. Note that ΔR, as given by (14), is positive if the cumulative reward from the back and forth movement (in the first $2M$ steps) is less than the reward in the last $2M$ steps. However, as T increases, the position of the last step approaches ξ_0 and it can be shown that the cumulative reward of the last $2M$ steps decreases. This property will be proved later, and it is due to the convexity and monotonicity properties of f. When T further increases, some of the very last $2M$ steps of the greedy behavior enter the phase of oscillation around ξ_0, and for T sufficiently large, all $2M$ belong to the oscillation phase. Note, however, that the oscillation phase rewards are those closer to 1/2, which is the lower limit of all greedy steps. If the greedy algorithm is to be optimal, even the $2M$ oscillatory steps should bring a cumulative reward greater than the original back and forth movement. On the other hand, if we prove this last inequality, this will also prove (14), whose last $2M$ steps bring more reward than the $2M$ oscillatory steps.

Let $\left(\psi_1,\psi_2\right)$ be the pair of oscillation points around ξ_0, *i.e.* $\psi_1=x_1+\left\lfloor\left(\xi_0-x_1\right)/\delta\right\rfloor\delta$ and $\psi_2=\delta+\psi_1$. In the worst case, which has just been mentioned,

$$\Delta R=M\left(\frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega}+1-\frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}\right)-\left[M+f\left(x_1\right)-f\left(x_1+M\delta\right)\right]$$
$$=M\left(\frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega}-\frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}\right)-\left[f\left(x_1\right)-f\left(x_1+M\delta\right)\right].$$

However, $f\left(x_1\right)-f\left(x_1+M\delta\right)$ can be seen as the sum of M terms $f\left(x_1+i\delta\right)-f\left(x_1+\left(i+1\right)\delta\right)$, for $i=0$, $M-1$. We shall further prove that each of these terms is smaller than the difference inside the big parentheses, *i.e.*

$$f\left(x_1+i\delta\right)-f\left(x_1+\left(i+1\right)\delta\right)\leq\frac{\omega\beta^{\psi_1}}{\omega\beta^{\psi_1}+1-\omega}-\frac{\omega\beta^{\psi_2}}{\omega\beta^{\psi_2}+1-\omega}. \tag{14}$$

This is a consequence of the following lemma:

Lemma 1. *For any concave function* $f\left(x\right)$ *the following inequality is true*:

$$f\left(x\right)-f\left(x+\Delta x\right)\leq f\left(x+\Delta x\right)-f\left(x+2\Delta x\right). \tag{15}$$

Inequality (15) holds because

$$\frac{f(x)-f(x+\Delta x)}{\Delta x} \geq f'(x+\Delta x) \geq \frac{f(x+\Delta x)-f(x+2\Delta x)}{\Delta x} \tag{16}$$

which is a consequence of the mean value theorem stating that there is a point ϕ_1 in $(x, x+\Delta x)$ such that $f'(\phi_1) = \left[f(x+\Delta x)-f(x)\right]/\Delta x$. Also, there is a point ϕ_2 in $(x+\Delta x, x+2\Delta x)$ such that $f'(\phi_2) = \left[f(x+2\Delta x)-f(x+\Delta x)\right]/\Delta x$. However, f is a concave function, and its derivative is non-increasing, therefore $\phi_1 \leq x+\Delta x \leq \phi_2$ implies $f'(\phi_1) \geq f'(x+\Delta x) \geq f'(\phi_2)$, which proves (16). In fact (15) can be easily generalized to any same length intervals, even overlapping ones, *i.e.* if $x_1 \leq x_2$, then

$$f(x_1)-f(x_1+\Delta x) \leq f(x_2)-f(x_2+\Delta x). \tag{17}$$

Due to (15) each successive equal length (*i.e.* Δx) interval produces an incremental reward $f(x)-f(x+\Delta x)$, which is smaller than the incremental reward of the next interval, and of all succeeding intervals, as long as f remains concave. Effectively, Lemma 1 proves that the incremental reward of the rightmost interval, which does not contain ξ_0, *i.e.* the interval $(\psi_1-\delta, \psi_1)$, is the highest among the rewards of all intervals of the same length, which begin to the left of $\psi_1-\delta$. Unfortunately, our aim was to prove (14), which would be secured if f remained concave in ψ_1, ψ_2, e.g. if $\psi_1 = \xi_0-\delta$ and $\psi_2 = \xi_0$. However this is not true, since at ξ_0 f turns from concave to convex. Fortunately, the term $f(\psi_1)-f(\psi_2)$, which covers the interval (ψ_1, ψ_2) can be seen as the sum of rewards related with the concave f in (ψ_1, ξ_0) and the concave $1-f$ in (ξ_0, ψ_2). Due to the odd symmetry around ξ_0, $f(\xi_0+(\psi_2-\xi_0))+f(\xi_0-(\psi_2-\xi_0)) = 2f(\xi_0)$, therefore $f(\psi_2) = 2f(\xi_0)-f(2\xi_0-\psi_2)$, and $f(\psi_1)-f(\psi_2) = f(\psi_1)-\left[2f(\xi_0)-f(2\xi_0-\psi_2)\right] = \left[f(\psi_1)-f(\xi_0)\right]+\left[f(2\xi_0-\psi_2)-f(\xi_0)\right]$.

However, due to the concavity of f, $f(\psi_1)-f(\xi_0) \geq f(\psi_1-\delta)-f(\xi_0-\delta)$, and $f(2\xi_0-\psi_2)-f(\xi_0) \geq f(2\xi_0-\psi_2-(\xi_0-\psi_1))-f(\xi_0-(\xi_0-\psi_1)) = f(\xi_0-\delta)-f(\psi_1)$. Therefore $f(\psi_1)-f(\psi_2) \geq \left[f(\psi_1-\delta)-f(\xi_0-\delta)\right]+\left[f(\xi_0-\delta)-f(\psi_1)\right] = f(\psi_1-\delta)-f(\psi_1)$.

This result states that the interval $(\psi_1, \psi_1+\delta)$, which contains ξ_0, provides higher Δf than the previous interval $(\psi_1-\delta, \psi_1)$, which in turn is higher than the Δf of any previous interval of the same length.

Therefore we have seen so far that a sequence of penalties, which begins at some $x < \xi_0$ and involves one fold, can be reduced to a sequence without any folds, and with improved total reward, as shown in **Figure 4**. In **Figure 4** a sequence of steps with a single fold, which starts at x_1 and ends at x_2, is shown together with the respective greedy sequence, which starts at x_1 and ends at $x_3 = 2M\delta + x_2$. If the sequence must extend after ξ_0, the additional steps are oscillation steps around ξ_0. The rest of this proof is just an application of this result, so that a sequence with an arbitrary number of folds can be reduced to an improved reward foldless sequence.

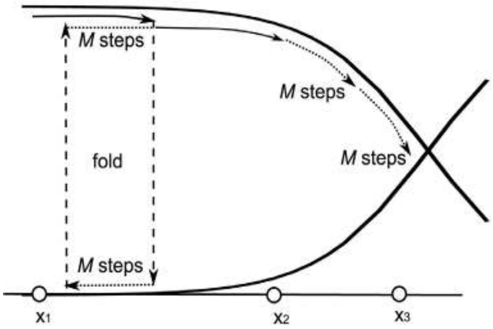


Figure 4. Reduction of a sequence of penalties, which contains a fold, to a sequence without folds and with improved total reward.

Suppose that the initial position of the game is at x_1, and that $x_1 \le \xi_0$ (otherwise reverse the initial probabilities $\omega, 1-\omega$). Suppose also that the initial sequence does not extend beyond ψ_2, *i.e.* it does not reach ξ_0 or it involves a number of oscillations around ξ_0. Then take the last fold and reduce it as mentioned, *i.e.* by replacing it with an equal number of greedy steps at the end of the current sequence. If these steps reach ξ_0, they are oscillation steps. Repeat the same step, until all folds have disappeared (oscillations do not count as folds). If the original sequence does extend beyond ξ_0, the approach is the same, but the reader should note that the application of this algorithm will finally reduce the part, which extends beyond ψ_2, to oscillations between ψ_1 and ψ_2.

6. Conclusion

We summarize the main results of this paper: An worst performance (adversarial) analysis of the Hedge algorithm has been presented, under the assumption of limited penalties per round. A recursive expression has been given for the evaluation of the maximum total cumulative loss; this expression can be exploited both numerically and analytically. However, binary penalty schemes provide an excellent approximation to the optimal scheme, and, remarkably, the greedy binary strategy has been proved optimal among binary schemes for the adversary.

References

[1] Robbins, H. (1952) Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematical Society*, **58**, 527-535. http://dx.doi.org/10.1090/S0002-9904-1952-09620-8

[2] Freund, Y. and Schapire, R.E. (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**, 119-139. http://dx.doi.org/10.1006/jcss.1997.1504

[3] Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R.E. (2002) The Non-Stochastic Multi-Armed Bandit Problem. *SIAM Journal on Computing*, **32**, 48-77. http://dx.doi.org/10.1137/S0097539701398375

[4] Allenberg-Neeman, C. and Neeman, B. (2004) Full Information Game with Gains and Losses. *Algorithmic Learning Theory*: 15*th International Conference*, **3244**, 264-278.

[5] Dani, V., Hayes, T.P. and Kakade, S.M. (2008) The Price of Bandit Information for Online Optimization. In: Platt, J.C., Koller, D., Singer, Y. and Roweis, S., Eds., *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, 345-352.

[6] Bartlett, P., Dani, V., Hayes, T., Kakade, S., Rakhlin, A. and Tewari, A. (2008) High-Probability Regret Bounds for Bandit Online Linear Optimization. *Proceedings of* 22*nd Annual Conference on Learning Theory* (COLT), Helsinki.

[7] Cesa-Bianchi, N. and Lugosi, G. (2012) Combinatorial Bandits. *Journal of Computer and System Sciences*, **78**, 1404-1422. http://dx.doi.org/10.1016/j.jcss.2012.01.001

[8] Uchiya, T., Nakamura, A. and Kudo, M. (2010) Algorithms for Adversarial Bandit Problems with Multiple Plays. In: Hutter, M., Stephan, F., Vovk, V. and Zeugmann, T., Eds., *Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence No. 6331, Springer, 375-389.

[9] Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R.E. (1995) Gambling in a Rigged Casino: The Adversarial Multi-Armed Bandit Problem. *Proceedings of* 36*th Annual Symposium on Foundations of Computer Science*, Milwaukee, 322-331.

[10] Hochbaum, D.S. (1995) Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, Boston.

[11] He, D., Chen, W., Wang, L. and Liu, T.-Y. (2013) Online Learning for Auction Mechanism in Bandit Setting. *Decision Support Systems*, **56**, 379-386. http://dx.doi.org/10.1016/j.dss.2013.07.004

[12] Park, C. and Lee, J. (2012) Intelligent Traffic Control Based on Multi-Armed Bandit and Wireless Scheduling Techniques. *International Conference on Advances in Vehicular System*, *Technologies and Applications*, Venice, 23-27.

[13] Bertsekas, D.P. (1998) Network Optimization. Athena Scientific, Belmont.

[14] Blum, A. and Burch, C. (2000) On-Line Learning and the Metrical Task System Problem. *Machine Learning*, **39**, 35-88. http://dx.doi.org/10.1023/A:1007621832648

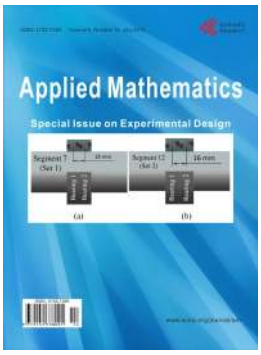
[15] Cole, S.J. and Lim, C. (2008) Algorithms for Network Interdiction and Fortification Games. *Springer Optimization and Its Applications*, **17**, 609-644. http://dx.doi.org/10.1007/978-0-387-77247-9_24

[16] Vaněk, O., Jakob, M. and Pěchouček, M. (2011) Using Agents to Improve International Maritime Transport Security. *IEEE Intelligent Systems*, **26**, 90-95. http://dx.doi.org/10.1109/MIS.2011.23

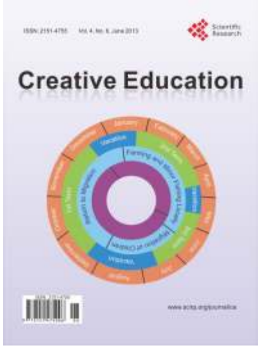
Advances in Bioscience and Biotechnology

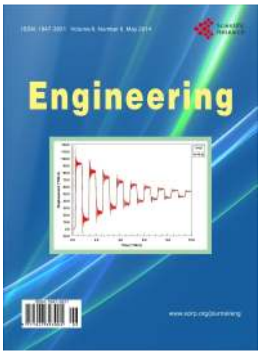
American Journal of Analytical Chemistry

American Journal of Plant Sciences

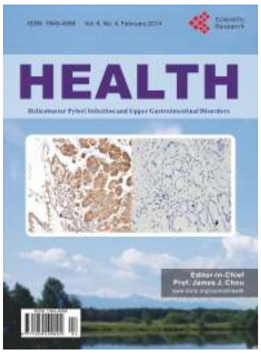

Applied Mathematics
Special Issue on Experimental Design

Agricultural Sciences
Special Issue on Research on Rice


Creative Education

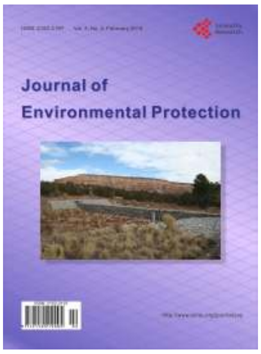

Engineering

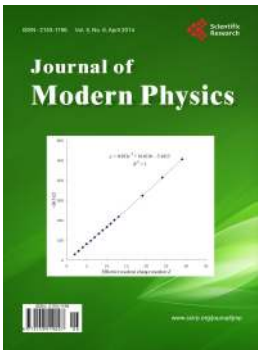
Food and Nutrition Sciences


HEALTH

Journal of Computer and Communications

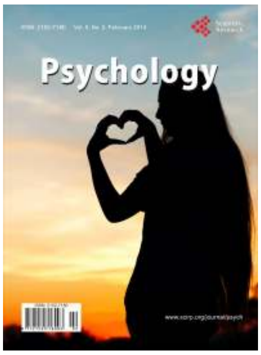
Journal of Cancer Therapy


Journal of Environmental Protection


Journal of Modern Physics

Modern Economy

Natural Science


Psychology