Aggregating Strategy for Online Auctions

Shigeaki Harada[1,⋆], Eiji Takimoto[2], and Akira Maruoka[2]

[1] NTT Service Integration Laboratories
harada.shigeaki@lab.ntt.co.jp
[2] Graduate School of Information Sciences, Tohoku University
Aoba 6-6-05, Aramaki, Sendai 980-8579, Japan
{t2, maruoka}@maruoka.ecei.tohoku.ac.jp

Abstract. We consider the online auction problem in which an auctioneer is selling an identical item each time when a new bidder arrives. It is known that results from online prediction can be applied and achieve a constant competitive ratio with respect to the best fixed price profit. These algorithms work on a predetermined set of price levels. We take into account the property that the rewards for the price levels are not independent and cast the problem as a more refined model of online prediction. We then use Vovk's Aggregating Strategy to derive a new algorithm. We give a general form of competitive ratio in terms of the price levels. The optimality of the Aggregating Strategy gives an evidence that our algorithm performs at least as well as the previously proposed ones.

1 Introduction

We consider the online auction problem proposed by Bar-Yossef, Hildrum, and Wu [3]. This models the situation where an auctioneer is selling single items in unlimited supply to bidders who arrive one at a time and each desires one copy. A particularly interesting case is for a digital good, of which infinitely many copies can be generated at no cost. Precisely, when each bidder t arrives with bid m_t, the auctioneer puts a price r_t on the item and sells a copy to the bidder at price r_t if $r_t \leq m_t$ and rejects the bidder otherwise. The auctioneer is required to compute the price r_t prior to knowing the values $m_t, m_{t+1}, \ldots$. Below we give a formal description.

Definition 1 (Online Auction A). *For each bidder* $t = 1, 2, \ldots, T$*,*

1. *Compute (randomly) a price* r_t*.*
2. *Observe the bid* $m_t > 0$*.*
3. *If* $r_t \leq m_t$*, then sell to bidder* t *at price* $g_{A,t} = r_t$*.*
4. *Otherwise, reject bidder* t *and* $g_{A,t} = 0$*.*

The total profit of the auction A *is* $G_{A,T} = \sum_{t=1}^{T} g_{A,t}$*.*

The goal of the auction is to make the total expected profit $E[G_{A,T}]$ as much as the best fixed price profit, denoted OPT, no matter what the bidding sequence is. Note that $\mathrm{OPT} = \max_{1 \leq k \leq T} k m_{(k)}$, where $m_{(k)}$ is the kth largest bid.

⋆ This work was done while the author was at Tohoku University.

D.Z. Chen and D.T. Lee (Eds.): COCOON 2006, LNCS 4112, pp. 33–41, 2006.

We first assume that the smallest value l and the largest values h of the bids in the auction are known. Discretizing the range $[l, h]$ with a finite set of price levels $h \geq b(1) > b(2) > \cdots > b(N) = l$, we have the problem reduced to an online prediction game with expert advice [4,5,8,9]. We use $b(i) = l\rho^{N-i}$ for some $\rho > 1$ with $N = O(\ln(h/l))$ so that $b(1) \geq h/\rho$. The idea is to introduce an expert for each price level $b(i)$ who always recommends the price $b(i)$. We can now use a number of expert-advice algorithms to achieve the total profit as much as that of the best expert, which is larger than OPT/ρ by the choice of the set of price levels. Blum, Kumar, Rudra, and Wu employ the Hedge or the Randomized Weighted Majority algorithm [9,5] and give a lower bound of

$$E[G_{\mathrm{Hedge},T}] \geq \frac{\ln \alpha}{\alpha - 1}\left(\frac{\mathrm{OPT}}{\rho} - \frac{h}{\ln \alpha}\ln(\log_\rho(h/l) + 1)\right)$$

on the total profit [1], where $\alpha > 1$ is a parameter of the Hedge algorithm. Blum and Hartline improve the additional loss term to $O(h)$ by using the Following Perturbed Leader (FPL) approach with a slight modification [2]. They call the modified version the Hallucinated-Gain (HG) algorithm and give the following bound

$$E[G_{\mathrm{HG},T}] \geq (1-\delta)\left(\frac{\mathrm{OPT}}{\rho} - 2h\left(\frac{2}{\delta}\ln \nu(\rho) + \frac{\nu(\rho)}{\delta^2}(1-\delta)^{\nu(\rho)} + 1\right)\right),$$

where $\nu(\rho) = \lfloor \log_\rho 2 \rfloor + 1$ and $\delta \in [0, 1]$ is a parameter of the HG algorithm. Moreover, the HG algorithm can be further improved so that it does not need to know l and h at a cost of only $O(h)$ additional loss.

In this paper, we first observe that, unlike the typical expert-advice setting, the rewards for the experts are not uniformly bounded. That is, the reward for expert i is either 0 or $b(i)$. So we could improve the algorithms using non-uniform risk information as in [7]. Furthermore, we have a further advantage in that the rewards for the experts are not independent. More precisely, when the bid m_t lies in $(b(i+1), b(i)]$, then all experts j with $j \geq i$ get rewards $b(j)$ and others get no rewards. In other words, there are only N possible outcomes to be considered. Taking this advantage into account, we give a more refined model of online prediction and apply Vovk's Aggregating Strategy [10] to derive a new algorithm called the Aggregating Algorithm for Auction (AAA). We give its profit bound[1] given by

$$E[G_{\mathrm{AAA},T}] \geq c(\alpha, B)\left(\frac{\mathrm{OPT}}{\rho} - \frac{1}{\ln \alpha}\ln(\log_\rho(h/l) + 1)\right),$$

where $\alpha > 1$ is a parameter of the AAA and $c(\alpha, B)$ is a complicated function of α and $B = \{b(1), \ldots, b(N)\}$. It seems that the bound is somewhat better since it has only an $O(\log\log(h/l))$ additional loss term, but in order to make $c(\alpha, B)$ a constant, we need to choose α that depends on h so that it quickly converges to

[1] Actually we obtain a tighter form of bound.

1 as h is large. Unfortunately, we have not succeeded to give a useful expression for $c(\alpha, B)$ to compare the profit bound with that of the HG algorithm, but it is better than the Hedge bound by the optimality of the Aggregating Strategy. We conjecture that the AAA performs as well as the HG algorithm. Numerical computation shows that the bound of the AAA outperforms others for sufficiently large ranges $[l, h]$ with $l = 1$ and $h \leq 10^{14}$.

2 Online Prediction Game and the Aggregating Strategy

We will show that online auction can be modeled as an online prediction game to which the Aggregating Strategy can be applied. The Aggregating Strategy is a very general method for designing algorithms that perform optimally for various games. In this section, we describe the strategy with its performance bound in a generic form.

First we describe a game that involves the learner (an algorithm), N experts, and the environment. A game is specified by a triple $(\Gamma, \Omega, \lambda)$, where Γ is a fixed prediction space, Ω is a fixed outcome space, and $\lambda : \Omega \times \Gamma \to [0, \infty]$ is a fixed reward function. (Note that the game is often described in terms of a loss function in the literature.) At each trial $t = 1, 2, \ldots, T$, the following happens.

1. Each expert i makes a prediction $x_{i,t} \in \Gamma$.
2. The learner combines $x_{i,t}$ and makes its own prediction $\gamma_t \in \Gamma$.
3. The environment chooses some outcome $\omega_t \in \Omega$.
4. The learner gets reward $\lambda(\omega_t, \gamma_t)$ and experts i get reward $\lambda(\omega_t, x_{i,t})$.

The total reward of the learner A is $R_{A,T} = \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t)$ and that of expert i is $R_{i,T} = \sum_{t=1}^{T} \lambda(\omega_t, x_{i,t})$. The goal of the learner A is to make predictions so that its total reward $R_{A,T}$ is not much less than the total reward of the best expert $\max_{1 \leq i \leq N} R_{i,T}$.

Now we give the Aggregating Strategy that derives an algorithm called the Aggregating Algorithm (AA) for each specific game. The AA uses a parameter $\alpha > 1$. For each trial t, the AA assigns to each expert i a weight $v_{i,t}$ given by

$$v_{i,t} = \frac{v_{i,1} \alpha^{R_{i,t-1}}}{\sum_{j=1}^{N} v_{j,1} \alpha^{R_{j,t-1}}}, \tag{1}$$

where $R_{i,t-1} = \sum_{q=1}^{t-1} \lambda(\omega_q, x_{i,q})$ is the sum of the rewards that expert i has received up to the previous trial. Initial weights $v_{i,1}$ can be set based on a prior confidence on the experts. Typically the uniform prior ($v_{i,1} = 1/N$) is used. When given predictions $x_{i,t}$ from experts, the AA predicts a $\gamma_t \in \Gamma$ given by

$$\gamma_t = \arg\sup_{\gamma \in \Gamma} \inf_{\omega \in \Omega} \frac{\lambda(\omega, \gamma)}{\log_\alpha \sum_{i=1}^{N} v_{i,t} \alpha^{\lambda(\omega, x_{i,t})}}. \tag{2}$$

The next theorem gives a performance bound of the AA.

Theorem 1 ([10]). *For any outcome sequence* $(\omega_1, \ldots, \omega_T) \in \Omega^*$,

$$R_{AA,T} \geq c(\alpha) \log_\alpha \sum_{i=1}^{N} v_{i,1} \alpha^{R_{i,T}} \geq c(\alpha) \max_{1 \leq i \leq N} \left(R_{i,T} - \frac{\ln(1/v_{1,i})}{\ln \alpha} \right),$$

where

$$c(\alpha) = \inf_{\boldsymbol{v},\boldsymbol{x}} \sup_{\gamma \in \Gamma} \inf_{\omega \in \Omega} \frac{\lambda(\omega, \gamma)}{\log_\alpha \sum_{i=1}^{N} v_i \alpha^{\lambda(\omega, x_i)}}, \tag{3}$$

where $\boldsymbol{v} = (v_1, \ldots, v_N)$ *ranges over all probability vectors of dimension* N *and* $\boldsymbol{x} = (x_1, \ldots, x_N)$ *ranges over all possible predictions of experts.*

3 The Game for Online Auction

Now we give the game $(\Gamma, \Omega, \lambda)$ reduced from the online auction problem. We first fix a finite set of price levels $B = \{b(1), \ldots, b(N)\}$ with $h \geq b(1) > \cdots > b(N) = l$ as options to choose from.

The prediction space Γ is the set of probability vectors of dimension N. The prediction $\gamma_t = \boldsymbol{p}_t = (p_t(1), \ldots, p_t(N)) \in \Gamma$ in the tth trial is interpreted as the way of choosing price r_t in the auction, i.e., letting $r_t = b(i)$ with probability $p_t(i)$. For each $1 \leq i \leq N$, we define an expert who always recommends the option $b(i)$. Formally, we let $x_{i,t} = \boldsymbol{e}_i (\in \Gamma)$, where $\boldsymbol{e}_i$ is the unit vector whose ith component is 1.

The outcome space Ω is the set of vectors whose ith component represents a reward for the ith option, which is either 0 (for the case where $m_t < b(i)$) or $b(i)$ (for the case where $m_t \geq b(i)$). Moreover, if the option $b(i)$ gets a positive reward, then all the options $b(j)$ with $j \geq i$ get positive rewards as well. Thus, we have only N possible reward vectors and

$$\Omega = \left\{ \begin{array}{c} (b(1), b(2), \ldots, b(N-1), b(N)), \\ (0, b(2), \ldots, b(N-1), b(N)), \\ \vdots \\ (0, 0, \ldots, 0, b(N)) \end{array} \right\}.$$

Let $\boldsymbol{b}_i = (0, \ldots, 0, b(i), \ldots, b(N))$ so that $\Omega = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N\}$. If the bid m_t lies in the interval $(b(i+1), b(i)]$ in the auction, then let the tth outcome be $\omega_t = \boldsymbol{b}_i$ in the reduced game.

Finally, our reward function is $\lambda(\boldsymbol{b}_i, \boldsymbol{p}) = \boldsymbol{b}_i \cdot \boldsymbol{p} = \sum_{j=i}^{N} b(j)p(j)$. Under the reduction just described, it is easy to see that $E[g_{A,t}] = \lambda(\boldsymbol{b}_i, \boldsymbol{p}_t)$ if the bid m_t is in $(b(i+1), b(i)]$, and so we have $E[G_{A,T}] = R_{A,T}$. Similarly, the total profit of a single sales price $b(i)$ equals $R_{i,T}$. So, Theorem 1 implies that the AA for the auction achieves profit nearly as large as the best single price sales $\max_i R_{i,T}$ in the set B. Moreover, if we choose $b(i) = l\rho^{N-i}$, then, no matter what the optimal price $r^* \in [l, h]$ is, there exists a $b(j)$ with $b(j) \le r^* < b(j+1) = \rho b(j)$ and so we have $\mathrm{OPT}/\rho \le R_{j,T} \le \max_i R_{i,T}$. Therefore, the AA achieves profit nearly as large as OPT. We call this algorithm the Aggregating Algorithm for Auction (AAA).

4 The Aggregating Algorithm for Auction

In this section we show how the AAA works by giving the weights $\boldsymbol{v}_t$ it maintains and the prediction $\boldsymbol{p}_t$ in a closed form. First we rewrite (1) and (2) in terms of the notations used in our auction game as

$$v_{i,t} = \frac{v_{i,1}\alpha^{b(i)\tau_{i,t-1}}}{\sum_{j=1}^N v_{j,1}\alpha^{b(j)\tau_{j,t-1}}}, \tag{4}$$

where $\tau_{i,t} = \#\{1 \le q \le t \mid m_t \le b(i)\}$ is the number of trials up to t in which the price $b(i)$ receives reward, and

$$\boldsymbol{p}_t = \arg\sup_{\boldsymbol{p}\in\Gamma} \min_{1\le k\le N} \frac{\boldsymbol{b}_k \cdot \boldsymbol{p}}{\log_\alpha\left(1 + \sum_{i=k}^N v_{i,t}(\alpha^{b(i)} - 1)\right)}. \tag{5}$$

Note that the Hedge algorithm predicts with $q_t(i) = v_{i,t}$ for determining the price at trial t. (More precisely, the normalized parameter $\alpha^{1/b(1)}$ is used instead of α in (4) [3].) The rest is to show the prediction of the AAA.

Theorem 2. *Let*

$$d_{k,t} = \log_\alpha\left(1 + \sum_{i=k}^N v_{i,t}(\alpha^{b(i)} - 1)\right)$$

for $1 \le k \le N$ *with the convention* $d_{N+1,t} = 0$. *Then,*

$$p_t(i) = \frac{\frac{1}{b(i)}(d_{i,t} - d_{i+1,t})}{\sum_{k=1}^N \frac{1}{b(k)}(d_{k,t} - d_{k+1,t})}$$

attains the supremum of (5).

Proof. Note that we want to solve

$$\boldsymbol{p}_t = \arg\sup_{\boldsymbol{p}\in\Gamma} \min_{1\le k\le N} \frac{\sum_{i=k}^N b(i)p(i)}{d_{k,t}}. \tag{6}$$

We first claim that for any $\boldsymbol{p} \in \Gamma$,

$$\min_{1\le k\le N} \frac{\sum_{i=k}^N b(i)p(i)}{d_{k,t}} \le \frac{1}{\sum_{k=1}^N \frac{1}{b(k)}(d_{k,t} - d_{k+1,t})}. \tag{7}$$

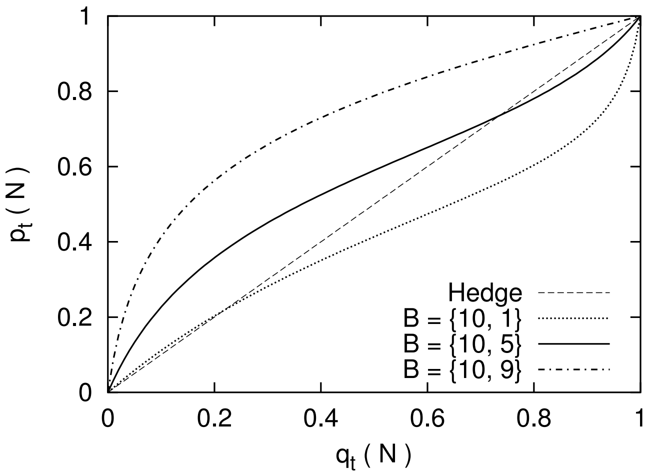

Fig. 1. The Hedge prediction $q_t(N)$ and the AAA prediction $p_t(N)$ for the lower price

Let M denote the r.h.s. of the above inequality. We prove the claim by contradiction. Assume on the contrary that the claim does not hold, i.e., for any $1 \leq k \leq N$, there exists a positive $\Delta_k > 0$ such that

$$\frac{\sum_{i=k}^{N} b(i)p(i)}{d_{k,t}} = M + \Delta_k.$$

Then we have

$$\begin{aligned} p(k) &= \frac{1}{b(k)}\big((M + \Delta_k)d_{k,t} - (M + \Delta_{k+1})d_{k+1,t}\big) \\ &= \frac{M(d_{k,t} - d_{k+1,t})}{b(k)} + \frac{\Delta_k d_{k,t}}{b(k)} - \frac{\Delta_{k+1} d_{k+1,t}}{b(k)} \\ &> \frac{M(d_{k,t} - d_{k+1,t})}{b(k)} + \frac{\Delta_k d_{k,t}}{b(k)} - \frac{\Delta_{k+1} d_{k+1,t}}{b(k+1)} \end{aligned}$$

since $b(k) > b(k+1)$. Summing up the both sides over all $1 \leq k \leq N$, we get

$$\sum_{k=1}^{N} p(k) > 1 + \frac{\Delta_1 d_{1,t}}{b(1)} > 1,$$

which contradicts the fact that $\boldsymbol{p}$ is a probability vector. So (7) holds.

On the other hand, the prediction $\boldsymbol{p} \in \Gamma$ with

$$p(i) = \frac{\frac{1}{b(i)}(d_{i,t} - d_{i+1,t})}{\sum_{k=1}^{N} \frac{1}{b(k)}(d_{k,t} - d_{k+1,t})}$$

clearly satisfies the equality of (7). This implies that this prediction $\boldsymbol{p}$ attains the supremum. □

The prediction of the AAA can be viewed as a nonlinear transformation of the Hedge prediction $q_t(i)$. Figure 1 illustrates the transformation for $N = 2$, $\alpha = 1.5$ and various sets of price levels $B = \{b(1), b(2)\}$. We fix $b(1) = 10$.

From the figure we can see that the AAA puts more weight on the lower price $b(N)$ when $q_t(N)$ is small. This is reasonable since the lower price is more likely to get reward. Curiously the weight on $b(N)$ gets larger when $b(N)$ gets closer to $b(1)$.

5 The Performance Bound of the AAA

In this section, we give the performance bound of the AAA by showing $c(\alpha)$ in terms of the set B of price levels. In what follows, we write $c(\alpha, B)$ to explicitly specify B. From the proof in Theorem 2, we can rewrite $c(\alpha)$ of (3) as

$$c(\alpha, B) = \inf_{\boldsymbol{v}\in\Gamma} \frac{1}{\sum_{k=1}^{N} \frac{1}{b(k)}(d_k - d_{k+1})}, \tag{8}$$

where

$$d_k = \log_\alpha \left(1 + \sum_{i=k}^{N} v(i)(\alpha^{b(i)} - 1)\right)$$

Theorem 3. *Let* $(r_1, \ldots, r_N)$ *and* $(s_1, \ldots, s_N)$ *be the probability vectors in* Γ *defined as*

$$r_i = \left(\frac{1}{b(i)} - \frac{1}{b(i-1)}\right) b(N),$$
$$s_i = \left(\frac{1}{\alpha^{b(i)} - 1} - \frac{1}{\alpha^{b(i-1)} - 1}\right)(\alpha^{b(N)} - 1)$$

with the convention that $b(0) = \infty$. *Then*

$$c(\alpha, B) = \frac{b(N)\ln\alpha}{D(\boldsymbol{r}||\boldsymbol{s}) + b(N)\ln\alpha}, \tag{9}$$

where $D(\boldsymbol{r}||\boldsymbol{s}) = \sum_{i=1}^{N} r_i \ln(r_i/s_i)$ *is the Kullback-Leibler divergence.*

Proof. The problem is to maximize the denominator of (8)

$$f(\boldsymbol{v}) = \sum_{k=1}^{N} \frac{1}{b(k)}(d_k - d_{k+1})$$

subject to $\boldsymbol{v} \in \Gamma$. First we relax the constraint and find the maximum of $f(\boldsymbol{v})$ subject to $\sum_{i=1}^{N} v(i) = 1$. Then we will show that the maximizer $\boldsymbol{v}^*$ lies in the feasible solution, i.e., $v^*(i) \geq 0$ for all i. Since f is concave, the set of equations

$$\frac{\partial}{\partial v(j)}\left(f(\boldsymbol{v}) + t\left(\sum_{i=1}^{N} v(i) - 1\right)\right)$$
$$= -(\alpha^{b(j)} - 1)\sum_{k=1}^{j}\left(\frac{1}{b(k)} - \frac{1}{b(k-1)}\right)\frac{1}{1 + \sum_{i=k}^{N}(\alpha^{b(i)} - 1)v(i)} + t = 0$$

for $1 \le j \le N$ and $\sum_{i=1}^{N} v(i) = 1$ give the maximizer. It is straightforward to show that the solution is

$$v^*(j) = \frac{F(b(j), b(j-1)) - F(b(j+1), b(j))}{t(\alpha^{b(j)} - 1)}$$

and

$$t = F(b(N+1), b(N)) = \frac{1/b(N)}{1 + 1/(\alpha^{b(N)} - 1)},$$

where $b(N+1) = -\infty$ and

$$F(x, y) = \frac{\frac{1}{x} - \frac{1}{y}}{\frac{1}{\alpha^x - 1} - \frac{1}{\alpha^y - 1}}.$$

We can show that $F(a, b) < F(b, c)$ for any $a < b < c$ with $b > 0$. This gives $v^*(j) > 0$.

Plugging $\boldsymbol{v}^*$ into $f(\boldsymbol{v})$, we have the theorem. □

6 Numerical Comparisons of the Performance Bounds

To compare the bound of the AAA with those of the Hedge and the HG algorithms, we need to give a useful form of $c(\alpha, B)$ with $b(i) = l\rho^{N-i}$ for $N = \lfloor \log_\rho(h/l) \rfloor + 1$. We have not succeeded to derive such an expression. So we show numerical experiments to compare the performance bounds. Recall that

$$E[G_{\mathrm{Hedge},T}] \ge \frac{\ln \alpha}{\alpha - 1} \frac{\mathrm{OPT}}{\rho} - \frac{h}{\alpha - 1} \ln\left(\log_\rho(h/l) + 1\right),$$

$$E[G_{\mathrm{HG},T}] \ge (1 - \delta)\frac{\mathrm{OPT}}{\rho} - 2h(1 - \delta)\left(\frac{2}{\delta}\ln \nu(\rho) + \frac{\nu(\rho)}{\delta^2}(1 - \delta)^{\nu(\rho)} + 1\right),$$

$$E[G_{\mathrm{AAA},T}] \ge c(\alpha, B)\frac{\mathrm{OPT}}{\rho} - \frac{c(\alpha, B)}{\ln \alpha} \ln\left(\lfloor \log_\rho h/l \rfloor + 1\right).$$

We fix $l = 1$ and adjust the parameters of the algorithms so that the first terms of the bounds are all equal to $(1/(2\rho))\mathrm{OPT}$. Thus, the bounds are all of the form of

$$E[G_{A,T}] \ge \frac{1}{2\rho}\mathrm{OPT} - g_A(h)h$$

for some functions g_A. Note that $g_{\mathrm{HG}}(h) = O(1)$ and $g_{\mathrm{Hedge}}(h) = O(\log \log h)$ by definition. Figure 2 shows how fast the functions $g_A(h)$ grow for the three algorithms.

Although g_{AAA} seems to be slightly increasing, the value is much smaller than g_{Hedge} and g_{HG} for a reasonable range of h. In fact, for a typical choice of $\rho = 1.01$, g_{HG} is a large constant (17.97) while $g_{\mathrm{AAA}} \le 0.5$ for $\log \log h \le 3.5$. It is interesting to note that the Hedge has a better bound than the HG bound in typical cases. We may improve the bound by using a tighter bound of Theorem 1 and choosing carefully the initial weights $v_{1,i}$.

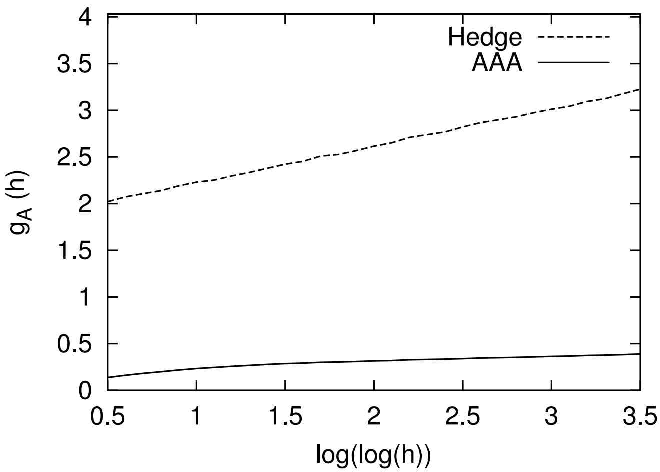

Fig. 2. The second term functions $g_A(h)$ for the three algorithms. We set $\rho = 1.01$. $g_{\mathrm{HG}}(h) = 17.97$.

References

1. A. Blum, V. Kumar, A. Rudra, and F. Wu. Online Learning in Online Auctions. *Proc. 14th SODA*, 2003.
2. A. Blum and J. D. Hartline. Near-Optimal Online Auctions. *Proc. 16th SODA*, 1156–1163, 2005.
3. Z. Bar-Yossef, K. Hildrum and F. Wu. Incentive-Compatible Online Auctions for Digital Goods. *Proc. SODA 2002*, 964–970, 2002.
4. N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold. R. E. Schapire, and M. K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.
5. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS*, 55(1):119–139, 1997.
6. M. Hutter and J. Poland. Prediction with expert advice by following the perturbed leader for general weights. *LNAI*, 3244, 279–293, 2004.
7. S. Harada, E. Takimoto and A. Maruoka. Online Allocation with Risk Information. *LNAI*, 3734, 343–355, 2005.
8. A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *LNAI*, 2777, 26–40, 2003.
9. N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inform. Comput.*, 108(2):212–261, 1994.
10. V. Vovk. A game of prediction with expert advice. *JCSS*, 56(2):153–173, 1998.