Approximate Methods for Sequential Decision Making Using Expert Advice *

Thomas H. Chung
Department of Electrical Engineering, Stanford University, thchung@isl.stanford.edu

We consider a game theoretic approach for sequentially choosing decisions by combining the suggestions of a fixed number of experts. Since the optimal decision making strategy is often computationally expensive, we present a methodology for obtaining approximate strategies with provably good performance. These methods are applicable to any decision problem with bounded payoff function, are computationally feasible, and arise naturally as approximations to the exact solution. We illustrate the ideas by applying our results to the problem of predicting a sequence of letters drawn from a finite alphabet with the goal being to minimize the number of mistakes made.

1 The Setup

The space of possible decisions, $\mathcal{D}$, and the space of possible outcomes, $\mathcal{Y}$, are both assumed to be compact subsets of Euclidean space. We sequentially interact with the system for a total of T units of time, where T must be finite and known in advance. If we choose the sequence of decisions $d^T = (d_1, d_2, \ldots, d_T) \in \mathcal{D}^T$ and we observe the sequence of system states $y^T \in \mathcal{Y}^T$, then our cumulative performance at time T is

$$\Phi_T = \sum_{t=1}^{T} \phi(d_t, y_t)$$

where $\phi : \mathcal{D} \times \mathcal{Y} \rightarrow \mathbb{R}$ is called the *payoff function* which must be continuous. We make the assumption[1] that the decision made does not influence the system we are observing. This assumption holds under a variety of contexts including sequential prediction and gambling. When this assumption holds, it is possible to choose our decisions at random and talk about our expected payoff for each possible system state. In this case, if we choose the sequence of probability distributions $\{\mathrm{P}_{D_t}\}_{t=1}^{T}$ on $\mathcal{D}$, and the sequence of system states turns out to be y^T, then our *expected performance* at time T is

$$\overline{\Phi}_T = \sum_{t=1}^{T} \mathrm{E}_{D_t}\left[\phi(D_t, y_t)\right]$$

At each time t, and prior to choosing our distribution P_{D_t}, we view the choices of N experts who also each choose a probability distribution $\mathrm{P}_{\mathcal{E}_{i,t}}$ for time t. The expected performance of the ith expert at time T is then

$$\overline{\Phi}_{i,T} = \sum_{t=1}^{T} \mathrm{E}_{\mathcal{E}_{i,t}}\left[\phi(\mathcal{E}_{i,t}, y_t)\right]$$

We wish to sequentially choose our distributions on $\mathcal{D}$ to perform nearly as well as the best expert, no matter who turns out to be the best expert in the end. We take a game theoretic viewpoint and minimize the *net loss*,

$$\nu = \max_i \overline{\Phi}_{i,T} - \overline{\Phi}_T$$

uniformly over all possible outcome sequences y^T and all possible experts. Specifically, we hypothesize the existence of an opponent whose goal is to maximize the net loss. The choice of expert opinions and the choice of system states are viewed as "moves" by this hostile opponent. The value of ν when both players play optimally is called the value of the game.

2 Applications

By choosing $\mathcal{D}$, $\mathcal{Y}$, and ϕ appropriately, we can find several applications that are special cases of this setup.

1. m-ary prediction.
   $\mathcal{D} = \mathcal{Y} = \{1, 2, \ldots, m\}$
   $\phi(d, y) = \mathrm{I}_{\{d=y\}}$

   Equivalently, the payoff matrix is given by

*This material was partially based upon work supported by a National Science Foundation Graduate Fellowship.

[1] This assumption corresponds to being able to view each unit of time as playing a two-player zero-sum game with imperfect information (i.e. where each player chooses their decision simultaneously).

COLT 94 - 7/94 New Brunswick, N..J. USA
© 1994 ACM 0-89791-655-7/94/0007..$3.50

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

This is the problem of predicting a sequence of letters drawn from an arbitrary finite alphabet, with the goal being to maximize the number of correct predictions. By interpreting the sequence appropriately, this can correspond to predicting m possible types of weather, the m possible ranges of a fluctuating stock relative, or the m possible classes associated with a pattern (our setup only works in a supervised environment). By subtracting a constant $1/m$ from the payoff matrix, this setup can also correspond to gambling at fair odds on a sequence of m letters using bounded bet size, as shown in ([4]) for the binary case.

2. The insurance game.
   $\mathcal{D} = \mathcal{Y} = \{0, 1\}$
   $\phi(d, y) = -a \ \mathrm{I}_{\{d=0\}} \ - \ b \ \mathrm{I}_{\{d=1, y=1\}}$
   Equivalently, $\begin{pmatrix} -a & -a \\ 0 & -b \end{pmatrix}$.

   Each year a farmer must choose whether or not to insure his crops. If he does, he pays a flat rate of a dollars per year. If he doesn't, he either pays nothing if nothing happens that year, or else $b > a$ dollars if some catastrophe strikes. (This problem has been analyzed by Vovk [7].)
3. Discrete time signal prediction.
   $\mathcal{D} = \mathcal{Y} = [-B, B]$
   $\phi(d, y) = -(y - d)^2$

   This is the problem of predicting a discrete signal one time unit in advance using squared error loss.
4. Gambling on a finite alphabet.
   $\mathcal{D} = \{(d_1, \dots, d_m) : d_i \geq 0, \sum_j d_j = 1\}$
   $\mathcal{Y} = \{1, \dots, m\}$
   $\phi(d, y) = \log(m d_y)$

   This is the problem of gambling on a sequence of letters drawn from a finite alphabet. Our decision at each time corresponds to what fraction of our wealth we place on each letter, and the wealth we receive is given at fair odds. Thus, the amount of money placed on the correct letter increases by a factor of m, while money placed on incorrect letters is lost. We measure payoff by log-wealth since log-wealth is additive and has a good interpretation as the wealth growth rate. The solution to this problem is given in [2].
5. Portfolio choice with bounded volatility.
   $\mathcal{D} = \{(d_1, \dots, d_m) : d_i \geq 0, \sum_j d_j = 1\}$
   $\mathcal{Y} = [\frac{1}{B}, B]^m$
   $\phi(d, y) = \log(d \cdot y)$

   This is the problem of how to choose a stock portfolio from among m stocks if we know in advance that each stock price cannot change by more than a factor of B in a single day. (We obtain the unbounded volatility case in the limit that $B \rightarrow \infty$.) Our decision at each time corresponds to the fraction of our wealth invested in each stock. The system state y is the observed vector of stock relatives where the ith component is the factor by which the price of the ith stock changes. Again, we measure payoff by log wealth, and wish to maximize our wealth growth rate.

Notice that an expert can correspond to any algorithm that yields a suggested probability distribution at each time, and need not be an actual human expert. Consequently, the set of "experts" might be made to include algorithms optimized for various modeling assumptions such as kth order Markov models. The set might also contain algorithms derived from various viewpoints such as the minimum description length principle or from Bayesian methods. The theory developed therefore not only shows how to combine expert advice, but also shows how to perform robust decision making and sheds light on the limits of how good a universal decision making algorithm can be.

3 Foundations

This setup has been considered in various forms by several authors, ([1], [4], [7]). A theory for optimal decision making in this context has been developed in [3], (see also [2]), and is based on a function V^* called the value function. This function gives the value of the net loss that will occur at the end of the game if both players play optimally from now on as a function of the current time index t and the past history of the game. One expression for V^* is given by the following equation taken from [3].

$$\begin{aligned} V^*(t; x_{1,t}, \dots, x_{N,t}) = \max_{\mathrm{P}_{Y_{t+1}^T}} \Bigg[\\ \max_{\{e_{i,s}(\cdot) \in \mathcal{D}\}} \mathrm{E}_{Y_{t+1}^T} \left[\max_i \left(x_{i,t} + \sum_{s=t+1}^{T} \phi(e_{i,s}(Y_{t+1}^{s-1}), Y_s) \right) \right] \\ - \max_{\{d_s(\,) \in \mathcal{D}\}} \mathrm{E}_{Y_{t+1}^T} \left[\sum_{s=t+1}^{T} \phi(d_s(Y_{t+1}^{s-1}), Y_s) \right] \Bigg] \qquad (1) \end{aligned}$$

where $x_{i,t} = \overline{\Phi}_{i,t} - \overline{\Phi}_t$ is the amount by which the ith expert is beating us at time t. Note that V^* only depends on the past history of the game through the set of differences, $\{x_{i,t}\}_{i=1}^N$. The function V^* plays a central role in the theory because the value of the game (our worst case net loss) is $V^*(0; 0^N)$, and furthermore, our optimal distribution at time t is given in terms of V^* as

$$\begin{aligned} \mathrm{P}_{D_t^*} \quad = \quad & \arg\min_{\mathrm{P}_{D_t}} \max_{y_t} \left(V^*(t; \{\overline{\Phi}_{i,t}(y^{t-1}, y_t)\}_{i=1}^N) \right. \\ & \left. - \mathrm{E}_{D_t} \left[\phi(D_t, y_t) \right] \right) \qquad (2) \end{aligned}$$

Note that this distribution is simply the optimal mixed strategy for the minimizing player in a two-player, zero-sum game whose payoff matrix is

$$f(y,d) = V^*(t;\{\overline{\Phi}_{i,t}(y^{t-1},y)\}_{i=1}^N) - \phi(d,y)$$

Thus, by using methods from the theory of games, it is possible to calculate this optimal distribution very efficiently if we can calculate $f(y,d)$ easily for any given arguments. Examples of such methods for finite games include the simplex algorithm and the Shapley-Snow methods (see [6]). In certain special cases (like the examples considered in section 6) it is possible to give a simple procedure for obtaining the optimal distribution from the function $f(y,d)$, (or equivalently, from V^*).

Unfortunately, only in rare circumstances does there exist a nice closed form for V^*. Whenever $\mathcal{D}$ and $\mathcal{Y}$ are finite, it turns out to be possible to calculate V^* exactly in polynomial time using a dynamic programming argument (as shown in [3]), but the degree of the polynomial grows with N, $|\mathcal{D}|$, and $|\mathcal{Y}|$ so that this method is not usually practical. Let us therefore suppose that we cannot calculate V^* exactly but are able to obtain an approximation, V, by some easily computable method. We might then consider choosing our distributions to satisfy the following condition.

The $D^*(V)$ algorithm:
At time t, choose a distribution $\mathrm{P}_{\hat{D}_t^*}$ satisfying

$$\begin{aligned}\mathrm{P}_{\hat{D}_t^*} \;=\; & \arg\min_{\mathrm{P}_{D_t}}\max_{y_t}\big(V(t;\{\overline{\Phi}_{i,t}(y^{t-1},y_t)\}_{i=1}^N) \\ & \qquad - \mathrm{E}_{D_t}[\phi(D_t,y_t)]\big)\end{aligned}$$

This method is the obvious choice because it coincides with the optimal strategy, (as given by (2)), when V equals V^*. As before, calculating the distribution $\mathrm{P}_{\hat{D}_t^*}$ that satisfies this condition is straightforward if the function V can be easily calculated for any set of arguments.

The basic result is given by the following theorem. If we use a "pessimistic" approximation to the value function, then the actual value of the net loss when using a strategy based on this approximation is never worse than our pessimistic approximation for the value of the game. Such a function is called an *overvalue* function.

Definition: A function, $V : \{0,1,\ldots,T\} \times \mathrm{I\!R}^{\mathrm{N}} \to \mathrm{I\!R}$, is called an *overvalue function* if it satisfies the following two conditions.

(*i*) $V(T;x^N) \geq \max_i x_i$
for all $x^N \in \mathrm{I\!R}^{\mathrm{N}}$,

(*ii*) $V(t-1;x^N) \geq \max_{\{\mathrm{P}_{\mathcal{E}_i}\}_{i=1}^N} \min_{\mathrm{P}_D} \max_{y\in\mathcal{Y}} \Big[V(t;\{x_i + \mathrm{E}_{\mathcal{E}_i}[\phi(\mathcal{E}_i,y)]\}_{i=1}^N) - \mathrm{E}_D[\phi(D,y)] \Big]$
for all $x^N \in \mathrm{I\!R}^{\mathrm{N}}$.

The value function V^* satisfies these conditions with equality, which is why we refer to functions that satisfy (i) and (ii) as overvalue functions; they consistently overestimate the value function. Note, however, that a function that always overestimates V^* need not be an overvalue function since this is not sufficient to guarantee (ii).

Theorem 1 (Overvalue Theorem)

Let V be an overvalue function. If we always choose our distribution according to the $D^*(V)$ algorithm, then the net loss is bounded above by $V(0;0^N)$, i.e.,

$$\max_i \overline{\Phi}_{i,T} - \overline{\Phi}_T \leq V(0;0^N)$$

for every sequence of advice $\{\{\mathrm{P}_{\mathcal{E}_{i,t}}\}_{t=1}^T\}_{i=1}^N$ and every sequence $y^T \in \mathcal{Y}^T$.

Thus, given any overvalue function V, we immediately have a bound on the worst case net loss when using the $D^*(V)$ algorithm.

4 Finding Overvalue Functions

The Overvalue Theorem shows that any overvalue function will yield a strategy with provably good performance, and furthermore, the best performance will be achieved by the overvalue function that overestimates V^* by as little as possible. (Ideally, we should use V^* itself.) Since every overvalue function must overestimate V^*, a good way to look for overvalue functions is to take the expression for V^* given in (1), find the tightest possible upper bound for each set of arguments, and hope that these bounds satisfy the overvalue conditions. In practice, this method works quite well, and any method for overestimating V^* will typically yield an overvalue function.

To upper bound V^*, we find it useful to make the following two definitions.

Definition: A class, $\mathcal{C}$, of distributions on $\mathcal{Y}$ is called *complete* if, for every continuous function, $g : \mathcal{Y} \to \mathrm{I\!R}$, $\mathcal{C}$ contains a distribution P_{Y_g} satisfying

$$\mathrm{P}_{Y_g} = \arg\max_{\mathrm{P}_Y}\min_d \mathrm{E}_Y\left[g(Y) - \phi(d,Y)\right]$$

Definition: Given any $\mathcal{L}^1$ random variable X, we define its upper α expectation for each $\alpha \in [0,1]$ by

$$\mathrm{U}[\alpha;X] = \int_{1-\alpha}^{1} F^{-1}(u)\mathrm{d}u$$

where F^{-1} is the quantile function of X, i.e. the left continuous pseudo-inverse of the cdf of X. (Note that U is a function of the *distribution* of X.)

Using these two definitions, it is possible to write down an explicit formula for an overvalue function that will turn out to be very useful.

Theorem 2 *Let $\mathcal{C}$ be complete, and define $X(e,Y)$ to be the random variable given by*

$$X(e,Y) = \phi(e,Y) - \max_{d\in\mathcal{D}} \mathrm{E}_Y\left[\phi(d,Y)\right], \tag{3}$$

for each $e \in \mathcal{D}$ and $\mathrm{P}_Y \in \mathcal{C}$. If A is a random variable satisfying

$$\mathrm{U}\left[\alpha; A\right] \geq \max_{\mathrm{P}_Y\in\mathcal{C}} \max_{e\in\mathcal{D}} \mathrm{U}\left[\alpha; X(e,Y)\right] \tag{4}$$

for all $\alpha \in [0,1]$, then

$$V_A(t;x^N) = \max_{\alpha_i\in\mathbb{P}_N} \sum_{i=1}^{N} \mathrm{U}\left[\alpha_i; x_i + \sum_{s=t+1}^{T} A_s\right] \tag{5}$$

is an overvalue function where $\mathbb{P}_N$ is the N dimensional probability simplex and $\{A_s\}_{s=1}^{\infty}$ are i.i.d. random variables with the same distribution as A.

At first glance, this theorem looks intimidating because of the seemingly complex conditions required on $\mathcal{C}$ and A, and because of the complex looking overvalue function, V_A, that we obtain. We will find, however, that the conditions on $\mathcal{C}$ and A are not difficult to verify in practice, and the resulting overvalue function V_A can be computed very efficiently. We will now elaborate on each of these issues.

We begin by considering the problem of finding a complete class $\mathcal{C}$. From the definition, it is clear that we could always take $\mathcal{C}$ to be the set of all possible distributions on $\mathcal{Y}$, but this set is usually much larger than is necessary, and it is desirable to have $\mathcal{C}$ be as small as possible in order to yield the tightest performance bounds. We will define a set $\mathcal{C}$ to be *minimally complete* if we cannot remove any distributions from $\mathcal{C}$ and still have a complete class. We will usually only be interested in classes of distributions that are minimally complete.

When $\mathcal{Y}$ is finite, the notion of completeness greatly simplifies because of the following result.

Theorem 3 *Let $\mathcal{Y}$ be finite, and let $\mathcal{C}$ be a set of distributions on $\mathcal{Y}$. Then $\mathcal{C}$ is complete if and only if for every subset $\mathcal{Y}_0 \subseteq \mathcal{Y}$, $\mathcal{C}$ contains a distribution P_{Y^*} satisfying*

$$\mathrm{P}_{Y^*} = \arg \min_{\mathrm{P}_Y\in\mathbb{P}_{\mathcal{Y}_0}} \max_{d} \mathrm{E}_Y\left[\phi(d,Y)\right]$$

where $\mathbb{P}_{\mathcal{Y}_0}$ is the set of distributions on $\mathcal{Y}_0$ and P_{Y^*} is required to be in $\mathbb{P}_{\mathcal{Y}_0}$.

This is a much simpler condition than given in the definition of completeness because it removes the dependence on the arbitrary function g. For each subset $\mathcal{Y}_0$ of $\mathcal{Y}$, a distribution satisfying the condition in the theorem is obtained by finding a solution to the two-player, zero-sum game with payoff matrix $\phi(d,y)$ when the minimizing player's class of decisions is restricted to $\mathcal{Y}_0$. As before, this can be done using any of various methods including the simplex algorithm and the Shapley-Snow method. Since $\mathcal{Y}$ has only finitely many subsets, the theorem shows that $\mathcal{C}$ can always be taken to be finite, and it also gives a method for finding $\mathcal{C}$; we simply solve this equation for each subset of $\mathcal{Y}$ and include the solution in $\mathcal{C}$. A minimally complete class can than be obtained from $\mathcal{C}$ by throwing out any unnecessary distributions.

We now consider how to find a random variable A that satisfies (4) once a suitable $\mathcal{C}$ has been found. This can be done quite easily in practice by making use of graphical properties of the U operator. To understand why this is so, first consider $\mathrm{U}\left[\alpha; X\right]$ as a function of α for a fixed X. It is a concave function on [0,1] and satisfies

$$\mathrm{U}\left[0; X\right] = 0$$
$$\mathrm{U}\left[1; X\right] = \mathrm{E}\left[X\right].$$

Now, for each $\mathrm{P}_Y \in \mathcal{C}$ and each $e \in \mathcal{D}$, the random variable $X(e,Y)$ (defined by (3)) will have mean less than or equal to zero. Consequently, the graph of $\mathrm{U}\left[\alpha; X(e,Y)\right]$ will be a concave function on [0,1] satisfying

$$\mathrm{U}\left[0; X(e,Y)\right] = 0$$
$$\mathrm{U}\left[1; X(e,Y)\right] \leq 0. \tag{6}$$

Equation (4) simply requires that the graph of $\mathrm{U}\left[\alpha; A\right]$ lie above the graph of $\mathrm{U}\left[\alpha; X(e,Y)\right]$ for each $\mathrm{P}_Y \in \mathcal{C}$ and each $e \in \mathcal{D}$. Now, given *any* continuous, concave function $h(\alpha)$ on [0,1] such that $h(0) = 0$, there exists a random variable A such that $\mathrm{U}\left[\alpha; A\right] = h(\alpha)$. Thus to find a random variable A satisfying (4), we simply find a concave function $h(\alpha)$ that lies above $\mathrm{U}\left[\alpha; X(e,Y)\right]$ for each $\mathrm{P}_Y \in \mathcal{C}$ and each $e \in \mathcal{D}$, and then find the random variable A that corresponds to h.

If $\mathcal{C}$ and $\mathcal{D}$ are finite, or else if ϕ is bounded, then there will always be a random variable, A, satisfying (4). In fact, from the previous discussion, it is clear that there will be infinitely many such random variables since we can always find a concave function that lies above another given concave function on [0,1]. Which random variable should we use? Looking back at the overvalue theorem, we see that our performance bound will be smallest when V_A is as small as possible. This turns out to be equivalent to choosing the random variable A^* whose U graph is as low as possible subject to the constraint (4). Thus, we should choose A^* so that $\mathrm{U}\left[\alpha; A^*\right]$ equals the convex hull of the graphs of $\mathrm{U}\left[\alpha; X(e,Y)\right]$. It is then clear that A^* will satisfy (4) in the tightest possible fashion, i.e., the graph of $\mathrm{U}\left[\alpha; A^*\right]$ will lie below that of $\mathrm{U}\left[\alpha; A\right]$ for every other A that satisfies (4). In general, A^* will depend on the particular choice of complete class $\mathcal{C}$.

Definition: The random variable A^* that satisfies (4) in the tightest possible fashion for a given set $\mathcal{C}$ is called the *minimal dilation* associated with $\mathcal{C}$.

As mentioned previously, a sufficient condition to guarantee that A^* exist is that the payoff function, ϕ, be bounded. Also, because of (6), A^* will *always* be a random variable with mean 0.

Finally, having found a distribution, A, satisfying (4), we consider how to calculate V_A for any values of its arguments. This can readily be accomplished using the following equivalent expression for V_A that can be derived from (5).

$$V_A(t; x^N) = (1-\beta)k + \sum_{i=1}^{N} \mathrm{E}\left[S_{i,t} \mathrm{I}_{\{S_{i,t}>k\}}\right] \quad (7)$$

where the various quantities are defined by

$$S_{i,t} = x_i + \sum_{s=t+1}^{T} A_s$$

$$\beta = \sum_i \mathrm{P}\{S_{i,t} > k\} \leq 1 \leq \sum_i \mathrm{P}\{S_{i,t} \geq k\} \quad (8)$$

Thus to calculate V_A, we first determine β and k that satisfy (8), and then calculate V_A from (7). In general, these steps can both be accomplished very quickly, the exact amount of computation depending on the distribution A. For the examples given in the next section, both of these steps can be accomplished in time $\mathrm{O}(T-t)$. This amount of time is quite reasonable and suggests that the $D^*(V_{A^*})$ algorithm offers an acceptable computational procedure for determining our distribution at each time.

If computation time is critically important, we might also consider other distributions satisfying (4). While these distributions won't yield performance as good as A^*, they may be easier to use in the sense that the complexity of $S_{i,t}$ will not grow as much with T. Since $S_{i,t}$ is obtained by adding independent and identically distributed (i.i.d.) copies of A, an obvious choice for A would be a normal random variable because then $S_{i,t}$ would also be normal. When ϕ is bounded, we can always find a normal distribution that satisfies (4), in which case V_A can be calculated in constant time (independent of T). The performance degradation that results from using the normal random variable rather than A^* is typically small (eg. the worst case net loss increases by no more than a factor of $\sqrt{\pi/2}$ for the examples in section 6).

We conclude this section by displaying some distributions that satisfy (4) for *any* bounded payoff function.

Theorem 4 *If the payoff function ϕ satisfies*

$$B^- \leq \phi(d, y) \leq B^+$$

for all $d \in \mathcal{D}$ and all $y \in \mathcal{Y}$, then (4) is satisfied by the scaled Bernoulli(1/2) random variable,

$$A = \begin{cases} (B^+ - B^-) & \textit{with prob. 1/2} \\ -(B^+ - B^-) & \textit{with prob. 1/2} \end{cases}$$

and also by a normal random variable with mean 0 and variance $\pi(B^+ - B^-)^2/2$.

The scaled Bernoulli random variable yields the better performance, but the normal random variable results in an overvalue function that is slightly easier to calculate. Thus, without any further work, we have found an A that works for any sequential decision problem posed in section 1 that has bounded payoff. The $D^*(V_A)$ algorithm yields a computationally feasible method with provably good performance $V_A(0; 0^N)$. Typically, the optimal A^* for a given problem will yield a net loss that is only a small factor better than the net loss achieved using either of the distributions given in theorem 4.

5 Performance Considerations

Having found a reasonable method for computing our distribution at each time, we might wonder how the performance behaves as a function of T and N, and also how this performance compares to the best possible performance that can be achieved. This section addresses these questions.

The Overvalue Theorem shows that the $D^*(V_A)$ algorithm achieves a net loss that is bounded by $V_A(0; 0^N)$. Since $V_A(0; 0^N)$ is computed from the distribution of the sum of T i.i.d. copies of A, it is not surprising that the normal distribution should play a role in the limit of large T. The following theorem gives the precise relationship.

Theorem 5 *Let A satisfy (4) and have mean 0 and variance σ_A^2. If f and F are the density and cdf of the standard normal distribution, respectively, then*

$$\begin{aligned} \lim_{T\to\infty} \frac{V_A(0;0^N)}{\sigma_A\sqrt{T}} &= Nf(F^{-1}(1-\frac{1}{N})) \\ &\approx \sqrt{2\ln N} \end{aligned}$$

where the approximation holds for large N.

Thus, in the limit of large T, the bound $V_A(0; 0^N)$ on the net loss is proportional to $\sqrt{T}$. This performance is quite good since we expect the performance of the best expert to grow linearly with T. Furthermore, in the limit of large N, the proportionality constant is approximately $\sigma_A\sqrt{2\ln N}$ which grows only minimally with N.

How does this performance compare to the best possible performance achievable? Under quite general conditions, it can be shown that $V^*(0; 0^N)$ must grow at least as quickly as $\Omega(\sqrt{T})$ for any N, and the proportionality constant must grow like $\Omega(\sqrt{\ln N})$ in the limit of large N. To prove this fact, we generalize a method proposed in [1]. Specifically, we consider experts that always suggest a degenerate distribution on $\mathcal{D}$, i.e. one that places all of its probability on a single decision in $\mathcal{D}$, where the particular decision suggested is chosen at random according to some distribution, $\mathrm{P}_{\mathcal{D}^*}$. The distribution $\mathrm{P}_{\mathcal{D}^*}$ that we use depends on the payoff function ϕ, and must be non-degenerate (i.e. not place all of its probability on a single decision). This method will only fail if the payoff function $\phi(d, y)$, when viewed as a two-player, zero-sum game, has an optimal pure strategy d^* for the maximizing player. Further details and extensions can be found in [3].

6 Example: Sequential Prediction

In this section, we illustrate the ideas by analyzing the problem of predicting a sequence of letters drawn from a finite alphabet with the goal being to maximize the number of correct predictions. (The $m = 2$ case has previously received considerable attention in [1].) We are thus considering the case,

$$\begin{aligned} \mathcal{Y} &= \{1,\ldots,m\} \\ \mathcal{D} &= \{1,\ldots,m\} \\ \phi(d,y) &= \begin{cases} 1 & \text{if y=d} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

This game can be interpreted in many ways (as discussed in section 2) depending on what each letter corresponds to.

We begin by finding a complete set of distributions. Theorem 3 is very easy to apply in this case because of the symmetry of the problem and yields the following corollary.

Corollary 1 *For the prediction problem, let $\mathcal{C}^*$ be the set of distributions on $\mathcal{Y}$ consisting of distributions that are uniformly distributed on some subset $\mathcal{Y}_0$ of $\mathcal{Y}$. Then $\mathcal{C}^*$ is minimally complete.*

We now find the distribution A^* that corresponds to the set $\mathcal{C}^*$ given in this corollary. To obtain A^*, we first plot $\mathrm{U}[\alpha; X(e,Y)]$ for each $\mathrm{P}_Y \in \mathcal{C}^*$ and each $e \in \mathcal{D}$. The $m = 4$ case is shown in figure 1 where we have only plotted those curves that are positive at some point. In general, there will only be $m - 1$ unique graphs. The graph of $\mathrm{U}[\alpha; A^*]$ is then obtained as the convex hull of these graphs as shown in figure 2. Finally, we can read off the distribution of A^* from its graph using a simple property of the U operator, namely

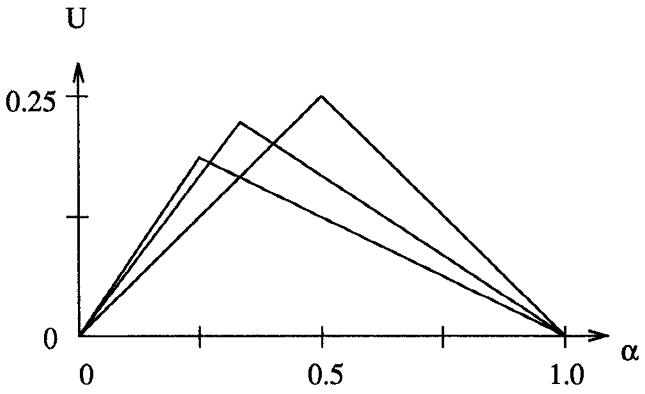

Figure 1: $\mathrm{U}[\alpha; X(e,Y)]$ graphs for $e \in \mathcal{D}$ and $\mathrm{P}_Y \in \mathcal{C}^*$.

$$\mathrm{P}\{A^* = a\} = \mathcal{L}\mathrm{eb}\{u : \frac{\mathrm{d}}{\mathrm{d}\tilde{\alpha}}\mathrm{U}[\tilde{\alpha}; A^*]\Big|_{\tilde{\alpha}=u} = a\}$$

Thus, the probability that A^* takes on the value a is simply the amount of time that the slope of $\mathrm{U}[\alpha; A^*]$ equals a. Using this fact, we easily obtain the distribution of A^* as given in the following theorem.

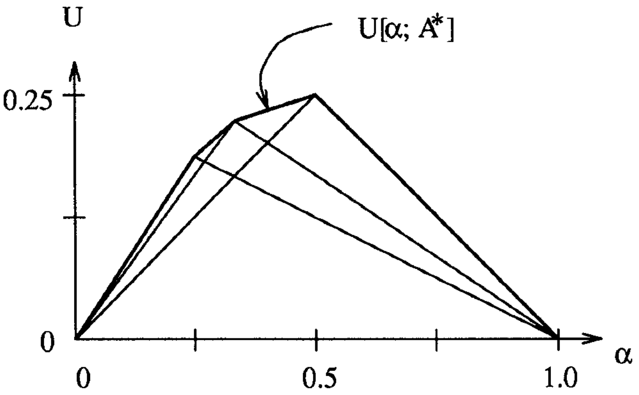

Figure 2: Obtaining the graph of $\mathrm{U}[\alpha; A^*]$ as the convex hull of the $\mathrm{U}[\alpha; X(e,Y)]$ graphs.

Theorem 6 *For the sequential prediction problem, the minimal dilation associated with $\mathcal{C}^*$ has the following distribution.*

$$A^* = \begin{cases} -\frac{1}{2} & \textit{with prob. } \frac{1}{2} \\ \frac{j^2-j-1}{j(j+1)} & \textit{with prob. } \frac{1}{j(j+1)} \quad j = 2,\ldots,m-1 \\ \frac{m-1}{m} & \textit{with prob. } \frac{1}{m} \end{cases}$$

It has mean 0 and variance $\sigma^2_{A^*} \leq \frac{7}{24}$ for all $m \geq 2$.

From theorem 5, we conclude that on a sequence of length T, we make about $\sigma_{A^*}\sqrt{2T\ln N}$ more mistakes than the best of N experts. This is negligible since we expect the number of errors for the best expert to be proportional to T. In the special case $m = 2$, we find that $\sigma_{A^*} = 1/2$ which agrees with the result given in [1]. For any given values of T and N, the value of V_{A^*} will typically be slightly better than the bound found in [1], but the difference will be negligible when N and T are both very large. Also note that $\sigma^2_{A^*}$ grows only minimally with m, which results in the net loss increasing by no more than about 8% as $m \to \infty$.

Having found A^*, we can also obtain other distributions satisfying (4). For example, each of the following distributions satisfies (4) for every m.

1. The uniform distribution on [-1,1].
2. The normal dist. with mean 0 and variance $\pi/8$.
3. $$A = \begin{cases} \frac{m-1}{m} & \text{with prob. } \frac{1}{2} \\ -\frac{m-1}{m} & \text{with prob. } \frac{1}{2} \end{cases}$$

For any given random variable, A, we compute the value of V_A from (7) and (8). When A is normal or binomial, this computation is somewhat simplified by noticing the following two facts (shown in [3]).

1. If Z is normal with mean 0 and variance 1, then
$$\mathrm{E}\left[Z\mathrm{I}_{\{Z>k\}}\right] = f(k)$$
where f is the density function,
$$f(z) = (1/\sqrt{2\pi})\exp(-z^2/2).$$

2. If $Z = \sum_{t=1}^{T} A_s$ where A_s are i.i.d. random variables with distribution,

$$A = \begin{cases} \frac{1}{2} & \text{with prob. } \frac{1}{2} \\ -\frac{1}{2} & \text{with prob. } \frac{1}{2} \end{cases}$$

then

$$\mathrm{E}\left[Z\mathrm{I}_{\{Z>k\}}\right] = \frac{T}{2^{T+1}} \binom{T-1}{\lfloor k + T/2 \rfloor}$$

Finally, we discuss how to compute the distribution suggested by the $D^*(V_A)$ algorithm now that V_A can be efficiently computed for any arguments. In general, to determine our distribution at time t, we must calculate $V_A(t; \{\overline{\Phi}_{i,t}(y^{t-1}, y)\})$ for each y in $\mathcal{Y}$. Here, $\overline{\Phi}_{i,t}(y^{t-1}, y)$, is the cumulative score of the ith expert up through time t if the next letter (the tth letter) turns out to be y. We can calculate each of these scores because we are given the distributions of each expert for time t prior to having to choose our own distribution at time t.

For the prediction problem, the optimal prediction can be computed by a "reverse water-filling" method. We begin by making a bar graph where we include a "bar" having width one and height $V_A(t; \{\overline{\Phi}_{i,t}(y^{t-1}, y)\}_{i=1}^{N})$ for each $y \in \mathcal{Y}$. This is shown in figure 3 for the $m = 4$ case. We reverse water-fill on this bar graph (gravity goes upward) with total water one. The amount of "water" that goes into each bar is precisely the amount of probability the distribution $\mathrm{P}_{\hat{D}_t^*}$ will allocate to that value of y.

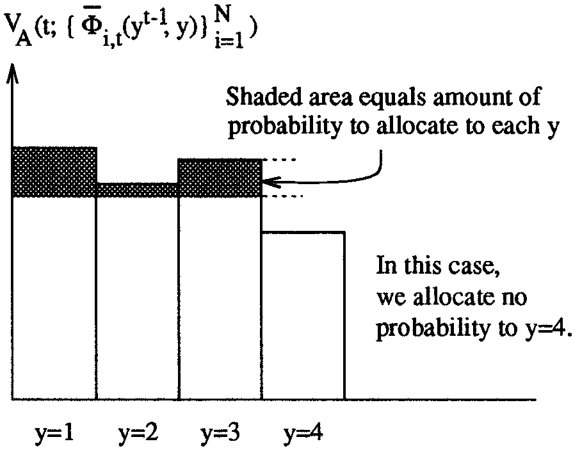
$V_A(t; \{\overline{\Phi}_{i,t}(y^{t-1}, y)\}_{i=1}^{N})$
Shaded area equals amount of probability to allocate to each y
In this case, we allocate no probability to y=4.
y=1
y=2
y=3
y=4

Figure 3: Obtaining our distribution by reverse water-filling.

References

[1] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. E. Schapire, and M. K. Warmuth. How to use expert advice. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 382–391, San Diego, May 1993.

[2] T. H. Chung. Sequential decision making using expert advice. To appear in the 1994 proceedings of the International Symposium on Information Theory.

[3] T. H. Chung. Prediction and iterated game theory via distributional majorization. Ph.D. Dissertation, Electrical Engineering, Stanford University, June 1994.

[4] T. M. Cover. Behavior of sequential predictors of binary sequences. In *Proceedings of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 263–272, 1965.

[5] J. Galambos. *The Asymptotic Theory of Extreme Order Statistics.* R. E. Kreiger, second edition, 1987.

[6] A. J. Jones. *Game Theory, Mathematical Models of Conflict.* Ellis Horwood Limited, Chichester, England, 1980.

[7] V. G. Vovk. Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann Publishers, 1990.