The Binning Algorithm and Continuous Experts

Efficiently computing the optimal strategy in various Predicting With Experts settings

Jacob Abernethy - TTI-C to UC Berkeley

Joint work with John Langford and Manfred Warmuth

The Online Learning Setting

- We must make some prediction y 2 [0,1] at every round t
- We are given access to a number of **experts** (a.k.a. prediction strategies, classifiers) whom we may ask for advice

The Online Learning Setting

Our Goal:

Perform well, *relative to the best expert*

What's a good strategy?

- Weighted Majority
- Follow the Leader
- Randomized Weighted Majority
- Binomial Weighting
- Follow the Perturbed Leader
- etc. ← Binning!

Are these optimal?

- Well, that depends…

Consider a Zero-Sum Game

- **The Master** vs **The Adversary**
- At every round:
  1. Adversary chooses $[x_{0t}, \ldots, x_{nt}]$ 2 $\{0,1\}^n$
  2. Master chooses $\hat{y}_t$ 2 $[0,1]$
  3. Adversary chooses y_t 2 $\{0,1\}$
  4. Payoff for Adversary is $|\hat{y}_t - y_t|$

How do we end the game?

1. Either, we restrict the game to T rounds, and T is known to both players

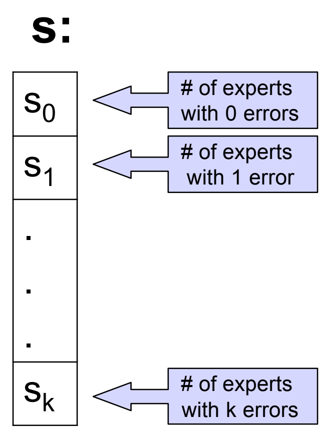
   (*How to use Expert Advice*, Cesa-Bianchi, Freund, Haussler, Helmbold, Schapire, Warmuth)
2. Or, we require that some expert may make no more than k mistakes, and k is known to both players

   (*On-line Prediction and Conversion Strategies,* Cesa-Bianchi, Freund, Helmbold, Warmuth)

The k-mistake rule

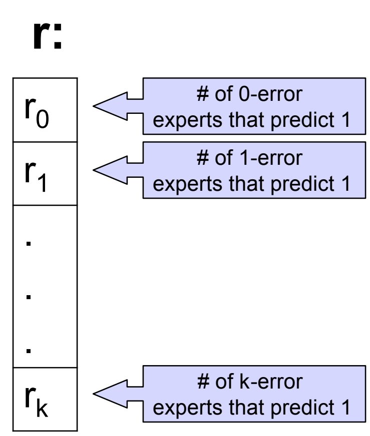
A New Representation

- After some number of rounds of the game, we represent our *state* by a (k+1)-dim vector **s**

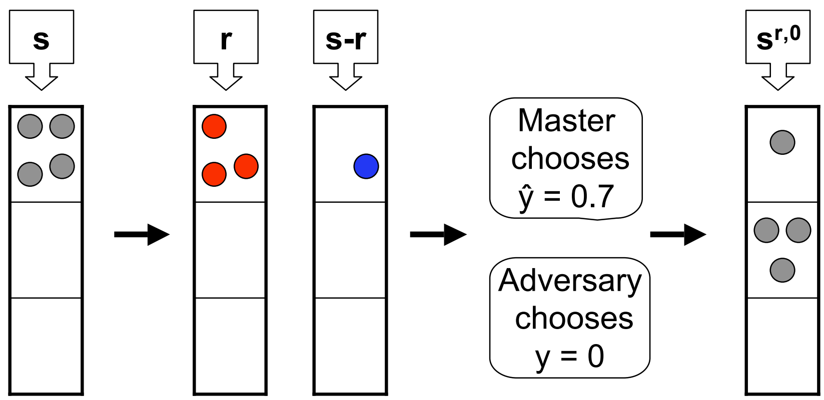


A new representation (part 2)

- On a given round, the experts predicting 1 can be represented by a k+1 dimensional *split* vector **r**
- Certainly **r** · **s**

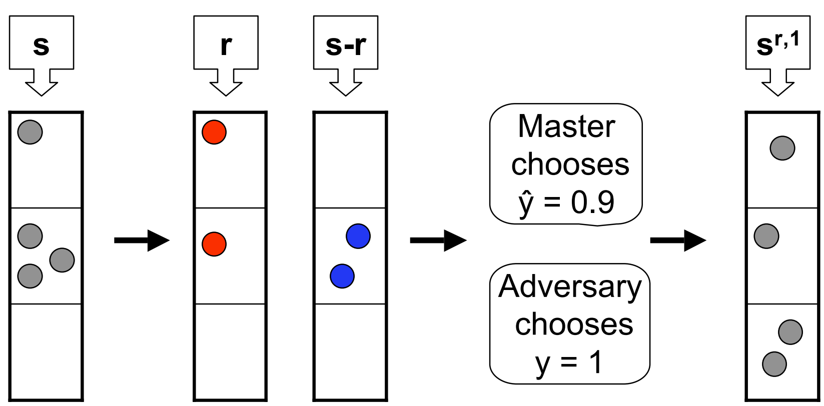


The Game in Action, k=2 and n=4



total loss = |0 - 0.7| = 0.7

The Game in Action, round 2



total loss = 0.7 + |1 - 0.9| = 0.8

A new representation (part 3)

Assume that we are in state **s,** we receive the split **r,** and we learn the outcome y. The next state can be defined by:

$$\mathbf{s}^{\mathbf{r},y} = \begin{cases} \mathbf{r}^{+} + (\mathbf{s}-\mathbf{r}) & \text{if } y=0 \\ \mathbf{r} + (\mathbf{s}-\mathbf{r})^{+} & \text{if } y=1 \end{cases}$$

where we define the + operator as:

$$s^{+} = (0, s_0, \ldots, s_{k-1}) \text{ when } s = (s_0, \ldots, s_k)$$

The value of the game

- The worst case (minimax) loss at state **s:**

$$\mathcal{L}(\mathbf{s}) := \begin{cases} -\infty & \text{if } \mathbf{s} = \mathbf{0} \\ 0 & \text{if } \mathbf{s} = (0,\ldots,0,1) \\ \max_{\mathbf{r}\le\mathbf{s}} \min_{\hat{y}\in[0,1]} \max_{y\in\{0,1\}} (|\hat{y}-y| + \mathcal{L}(\mathbf{s}^{\mathbf{r},y})) & \text{otherwise.} \end{cases}$$

- Which can be simplified to:

$$\mathcal{L}(\mathbf{s}) = \max_{\mathbf{r}\le\mathbf{s}} \left\{ \frac{\mathcal{L}(\mathbf{s}^{\mathbf{r},0}) + \mathcal{L}(\mathbf{s}^{\mathbf{r},1}) + 1}{2} \right\}$$

- But still hard to compute!

The optimal Master strategy

- If we can compute L then we can easily compute the optimal Master prediction strategy:

$$\hat{y} = \frac{\mathcal{L}(\mathbf{s}^{\mathbf{r},1}) - \mathcal{L}(\mathbf{s}^{\mathbf{r},0}) + 1}{2}$$

So can we compute L?

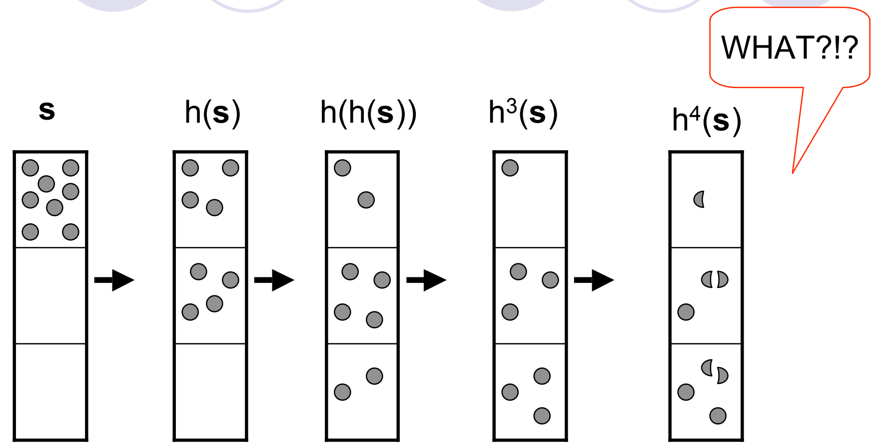
- Requires looking at all splits **r** · **s** …or does it?

No! A worst-case split is always the *middle* split, that is

$$\mathbf{r} = \mathbf{s}/2.$$

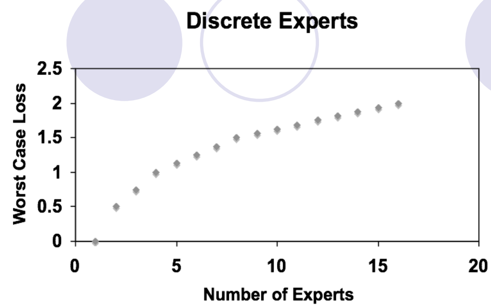
This follows by showing that L is concave!

When the game is played optimally:

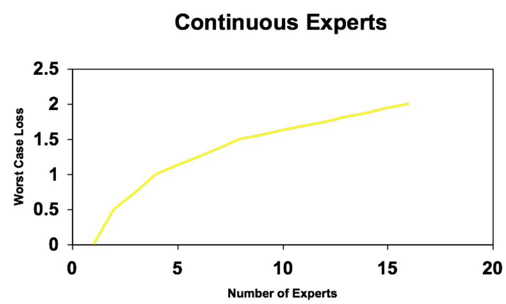


We need **Continuous** Experts!!

- Before: **s**,**r** 2 $\{0,\ldots,N\}^{k+1}$
  Now: **s**,**r** 2 $[0,N]^{k+1}$
- Simply gives MORE power to the adversary, i.e. more freedom to choose the split **r.**
- The difference: not much!



Consider an easy case: k = 0



Now L is easy to compute!

- Now we have a very simple way to compute L:

$$\mathcal{L}(\mathrm{s}) = \mathcal{L}(h(\mathrm{s})) + \frac{1}{2}$$

- Take the largest n such that $|h^n(\mathbf{s})|$ ¸ 1

$$\mathcal{L}(\mathrm{s}) = \frac{n}{2} + BaseLoss(h^n(\mathrm{s}))$$

- BaseLoss is defined to “smooth” out base cases, e.g. when only (1 + ε) experts left

Generalizes Binomial Weighting

- Binning Bound = How many times can I shift half of my expert weight down before I run out of experts?
- Binomial Weighting Bound = How many times can I halve the binomially-many virtual experts?
- These are the same thing!

A New Approach to Experts Algorithms

- Relaxed setting allows us to compute **optimal** strategy
- The same technique works more generally:
  1. Time-bounded game
  2. Experts' prediction in [0,1]
  3. The optimal Hedge algorithm