

Disclaimer: *These are rough notes, with some exercises.*

17.1 Applications?

Question: Two person games?

Let's return to two person games. Given a 0 – 1 payoff matrix M . The row player can play distributions over the rows and the column player can play distributions over the columns.

Question: And Von Neumann said?

The value, λ^* of the game is...

$$\lambda^* = \min_D \max_j M(D, j) = \max_P \min_i M(i, P).$$

where D is a distribution on rows and P is a distribution on columns. This is also duality; there are primal dual strategies that are equal.

Say, we actually don't know this, and instead define the following two concepts.

$$\lambda_r = \min_D \max_j M(D, j)$$

$$\lambda_c = \max_P \min_i M(i, P).$$

The row player is playing a distribution that lower bounds the value of the minimum column. The column player is playing a distribution that upper bounds the value of the minimum row.

Notice that weak duality says that $\lambda_r \geq \lambda_c$.

Question: Experts?

Let the experts be the rows. And the “events” or timesteps by a column that is played by an “adversary”; choose the best column for the column player for the current row distribution. (That is, the max over j in the definitions above.)

Question: What happens?

Well, we have the following bound on the cost of the experts situation.

$$C \leq (1 + \epsilon)m + \ln n/\epsilon.$$

That is, the total cost C is at most this function of the cost of the best row.

Define a row distribution, D_f to be the average of the distributions of the row player in this game over T steps, and a column distribution where each column is played in proportion to the number of times it was seen. We can translate the cost to be

$$M(D_f, P_f) = C/T \leq (1 + \epsilon)m/T + \ln n/(\epsilon T) = (1 + \epsilon)M(i, P_f) + \ln n/(\epsilon T)$$

where i is the best response to P_f . Setting $\epsilon = \delta/2$ and $T \geq 4 \ln n/\delta^2$ and noting that $D(i, P_f)$ is at most 1, we get

$$M(D_f, P_f) \leq \min_i M(i, P_f) + \delta$$

That is, the distribution D gets close to the value of the best response! We also have.

$$M(D_f, P_f) \leq \min_i M(i, P_f) + \delta \leq \lambda_c + \delta.$$

from the definition of λ_c is the highest value of a row response for the best column distribution which must be at least as good as P_f .

Question: What about P_f , does it do well against the best response on D_f ?

In each iteration, the column strategy received at least λ_r since it played the best column strategy against a current distribution which can be no worse for the column player than the best distribution in which it can get λ_r . Thus

$$M(D_f, P_f) \geq \lambda_r.$$

Thus for any row distribution, we have

$$\lambda_r \leq M(D_f, P_f) \leq \lambda_c + \delta.$$

This is strong duality (up to a δ .)

17.2 Learning.

Another setting, is the following.

Suppose that for any given a set of examples, there is an algorithm to classify the examples correctly with probability $1/2 + \gamma$. Indeed, suppose that for any distribution on the examples the algorithm can predict correctly with probability $1/2 + \gamma$. (This has been called a weak learner against any distribution.)

Can you learn with probability $1 - \epsilon$? on the examples?

Question: How do we set this up as an experts problem?

Start with N examples. And the uniform distribution. Make a hypothesis that learns the function with probability $1/2 + \gamma$ on this distribution.

Question: Then what?

Then, reweight the current distribution according to whether the hypothesis classified it correctly.

That is, if hypothesis was right, downweight the example.

Question: In each iteration, what is the expected penalty?

At least $1/2 + \gamma$ since the hypothesis is right on this fraction of the distribution.

Question: Final hypothesis?

Weighted average of all hypothesis.

Question: What is analysis?

S is incorrectly labelled examples. Penalty is at most $T/2$ since majority gets it wrong.

The potential function at time t , Φ^t is at most $\phi^1 e^{-\alpha T(1/2+\gamma)}$.

The potential function is at least $|S|(1-\alpha)^{T/2} \geq |S|e^{-\alpha+\alpha^2}T/2$.

Thus,

$$|S|/n \leq e^{-\alpha(\gamma-\alpha/2)T}.$$

Choosing $\alpha = \gamma$ and $T = \frac{2}{\gamma} \ln \frac{1}{\epsilon}$, we get that

$$|S|/n \leq \epsilon.$$

17.3 Arithmetic Coding.

Question: Given a sequence of characters, how can we code it?

Send it as a binary representation of a point in an interval.

Question: Huh?

Start with the interval $[0, 1)$. When sending t th character (chosen from a set Σ), divide the current interval into $|\Sigma|$ segments and pick the interval corresponding to the current character.

Question: How many bits?

If current interval is of size $\geq 2^{-m}$ we need only m bits to specify a point in the interval, this number uniquely specifies the string of t characters (given that you know how each subinterval is divided.)

Question: Wierd. So?

Well, lets say $\Sigma = \{0, 1\}$. And the probability of 0 is $1/3$ and the probability of 1 is $2/3$. We can then divide subintervals into $1/3$ and $2/3$. (Thus, 1111 is in interval of size $(2/3)^4 > (1/2)^3$. In can be coded by 111.)

Number of bits to code, c_1, \dots, c_T the size of the interval is $\prod p(c_i)$. The number of bits is $-\sum_i \log p(c_i)$.

The expected code length is

$$\sum_{c_1, \dots, c_T} p(c_1, \dots, c_T) [-\log p(c_1, \dots, c_T)].$$

Or approximately.. $H(p_T) + 1$. This is optimal in various senses according to Shannon.

Question: What if we don't know distribution?

Let's figure it out on-line. How. Set up an expert's situation over N prediction algorithms with "log-loss" since that is the length of the code.

Each prediction algorithm i has a probability distribution, p_i^t , for each character at time t .

Now, start with w_i on each algorithm which sum to 1. (E.g. $w_i = 1/N$.) Also, can be viewed as Bayesian prior.

Now, we update according to "log-loss".

Question: Huh?

Recall, at time t , we see character c_t , and prediction algorithm i , we reweight as follows.

$$w_i^{t+1} = w_i^t p_i^t(c_t),$$

where $p_i^t(c_t)$ is the probability that the i th prediction algorithm assigns to the character c_t that we see at time t .

Or, if we take the loss to be $L_i^t = -\log p_i^t(c_t)$, the rule is

$$w_i^{t+1} = w_i^t e^{-L_i^t}.$$

Question: What does the algorithm do?

Predict according to weighted distribution over prediction algorithms. Thus, we get that

$$p_A^t(c_t) = \frac{\sum_i w_i^t p_i^t(c_t)}{\sum_i w_i^t}.$$

And the log-loss is $-\log p_A^t(c_t)$.

Question: What is total weight at end?

Well, $W^t = \sum_i w_i^t$. Moreover,

$$\frac{W^{t+1}}{W^t} = \frac{\sum_i w_i^t e^{-L_i^t}}{\sum_i w_i^t} = \frac{\sum_i w_i^t p_i^t(c_t)}{\sum_i w_i^t} = p_A^t(c_t).$$

Taking logs, we get

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c_t).$$

The loss of the algorithm that step. Adding up (or multiplying up), we get

$$-\log \frac{W^T}{W_1} = -\sum_t \log p_A^t(c_t) = L_A^T$$

That is, we have that the negative log of the weight is the log-loss, L_A^T , of the algorithm. This is also the length of the code!!

Question: Do we do as well as the best expert?

Sure, we will set $w_i = 1/n$ initially. Now the total weight is at least the weight of the best expert. Thus, we have

$$\sum_i w_i^T \geq \frac{1}{n} \max_i w_i^T \geq \frac{1}{n} \max_i e^{-\sum_t L_i^t}$$

Taking negative logs, we get that

$$L_A^T \leq \log N + \min_i L_i^T.$$

That is we get code that is as good as your best expert to within an additive $\log N$ term.

Exercise 1: How do you decode? You need to now c_t to figure out how to partition the interval at c_{t+1} . How does this work? Am I making a mistake? Hmm.

Question: This is called on-line Bayesian learning?

The update rule is optimal according to bayes rule. It minimizes regret as well.