CSIE5002 Prediction, learning, and games

Lecture 5: Multiplicative weight update I

Yen-Huan Li (yenhuan.li@csie.ntu.edu.tw)

18.03.2019

Department of Computer Science and Information Engineering
National Taiwan University

Abstract

- Given a binary sequence $(\omega_1, \omega_2, \ldots, \omega_t) \in \{\,0, 1\,\}^t$, how do we predict the next bit x_{t+1}?

- What should be the performance measure?

- When can we claim a prediction algorithm *good* or even *optimal*?

Recommended reading

- N. Cesa-Bianchi and G. Lugosi. 2006. *Prediction, Learning, and Games.* Chapter 9.
- N. Littlestone and M. K. Warmuth. 1994. The weighted majority algorithm.
- R. El-Yaniv. 1998. Competitive solutions for online financial problems.
- A. DeSantis *et al.* 1988. Learning probabilistic prediction functions.
- T. Cover and J. A. Thomas. 2006. *Elements of Information Theory*. Chapter 5 & 13.

Table of contents

1. Weighted majority vote
2. Individual sequence prediction
3. Application: Universal source coding
4. Conclusions

Weighted majority vote

Problem formulation (1/2)

Suppose a group of people has to answer a binary question everyday, *without any prior knowledge of the correct answers*, how do they aggregate their opinions to yield the smallest number of mistakes?

Question. What if there is only one day (round)?

Exercise. Find a real-world scenario equivalent to this problem.

Problem formulation (2/2)

Protocol. (Voting) Define the initial loss $L_0 = 0$. For $t = 1, \ldots, T$, the following happen in order.

1. $\textsc{Expert}$ i announces $\gamma_t(i) \in \{0, 1\}$, $1 \le i \le m$.
2. $\textsc{Learner}$ announces $\gamma_t \in \{0, 1\}$.
3. $\textsc{Reality}$ announces $\omega_t \in \{0, 1\}$.
4. Update the cumulative loss: $L_t \leftarrow L_{t-1} + \lambda(\omega_t, \gamma_t)$, where

$$\lambda(\omega, \gamma) \coloneqq \mathbb{1}_{\{\omega \neq \gamma\}}, \quad \forall (\omega, \gamma) \in \{0, 1\}^2 .$$

Terminology. We say that the learner adopts an *online algorithm*.

Question. Is it reasonable to consider minimizing L_T?

Competitive analysis

Observation. In the worst case, it holds that $L_T = T$.

Definition. (Competitiveness) We say that an online algorithm is *β-competitive* for some $\beta \geq 0$ in the voting protocol, if and only if there exists some $\alpha \geq 0$ such that

$$\sum_{t=1}^{T} \lambda(\omega_t, \gamma_t) \leq \beta \min_{1 \leq i \leq m} \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t(i)) + \alpha.$$

The smallest such β is called the algorithm's *competitive ratio*.

A. Borodin and R. El-Yaniv. 1998. *Online Computation and Competitive Analysis*.

Halving algorithm

Proposition 1. Suppose in the voting protocol, there exists an expert who always outputs the correct answers. Then, there is an 1-competitive online algorithm.

Proof. Consider the *halving* algorithm. Let $\mathcal{V} = \{1, \ldots, m\}$. In each round, the algorithm outputs the result of majority vote in $\mathcal{V}$; after seeing ω_t, the algorithm removes the wrong experts from the set $\mathcal{V}$. Notice that whenever a mistake occurs, at least half of the experts are removed. Then, we have $L_T \leq \log_2 m$.

Halving algorithm with resets

Proposition 2. There is an $(1 + \log_2 m)$-competitive algorithm for the voting protocol.

Proof. Modify the halving algorithm such that if the set $\mathcal{V}$ becomes empty, then the algorithm sets $\mathcal{V} = \{ 1, \ldots, m \}$ for the next round. Define

$$L_T^\star := \min_{1 \le i \le m} \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t(i)).$$

Notice there can be at most $L_T^\star$ resets, and there are at most $(1 + \log_2 m)$ rounds between two consecutive resets. Then, we have

$$L_T \le (1 + \log_2 n) L_T^\star + \log_2 m.$$

Weighted majority algorithm

Theorem 1. There is a 2.41-competitive algorithm for the voting protocol.

Algorithm. (Weighted majority vote)

- Set $w(i) = 1$ for $1 \leq i \leq m$.
- For each t, the algorithm outputs $\gamma_t = 0$ if

$$\sum_{i:\gamma_t(i)=0} w(i) \geq \sum_{i:\gamma_t(i)=1} w(i),$$

  and outputs $\gamma_t = 1$ otherwise. Then, for each i, the algorithm halves $w(i)$ if the $\textsc{Expert}$ i makes a mistake.

N. Littlestone and M. K. Warmuth. 1994. The weighted majority algorithm.

V. G. Vovk. 1992. Universal forecasting algorithms.

Proof of Theorem 1

Proof. (Theorem 1) Notice that after T rounds, we have (why?)

$$\sum_{i=1}^{m} w(i) \leq \left(\frac{3}{4}\right)^{L_T} m.$$

Moreover, we have

$$\left(\frac{1}{2}\right)^{L_T^\star} \leq \sum_{i=1}^{m} w(i).$$

Combining the two inequalities, we obtain

$$-L_T^\star \leq L_T \log_2\left(\frac{3}{4}\right) + \log_2 m.$$

Individual sequence prediction

Problem formulation (1/3)

Consider the *alphabet* $\mathcal{A} = \{ 1, \ldots, m \}$ for some positive integer m. Given a *string* $(\omega_1, \ldots, \omega_t) \in \mathcal{A}^t$, what is the *conditional probability distribution* of the next symbol ω_{t+1}?

Remark. Notice that indeed, we do not assume a probabilistic model for the string.

Problem formulation (2/3)

Denote by Δ the probability simplex in $\mathbb{R}^m$, i.e.,

$$\Delta := \left\{ (x(1), \ldots, x(m)) \in \mathbb{R}^m \;\middle|\; x_i \geq 0 \; \forall 1 \leq i \leq m, \sum_i x_i = 1 \right\}.$$

Protocol. (Individual sequence prediction) Define the initial loss $L_0 = 0$. For $t = 1, \ldots, T$, the following happen in order.

1. $\textsc{Learner}$ announces $\gamma_t \in \Delta$.
2. $\textsc{Reality}$ announces $\omega_t \in \mathcal{A}$.
3. Update the cumulative loss: $L_t \leftarrow L_{t-1} + \lambda(\omega_t, \gamma_t)$, where

$$\lambda(\omega, \gamma) := -\log \gamma_t(\omega_t), \quad \forall (\omega, \gamma) \in \{ 1, \ldots, m \} \times \Delta.$$

Problem formulation (3/3)

Let $\mathcal{H}$ be a hypothesis class of functions $h : \mathcal{A}^* \to \Delta$.

Suppose that the LEARNER's predictions are given by

$$\gamma_t = \hat{p}(\omega_{1:t-1}) \in \Delta, \quad \forall 1 \leq t \leq T,$$

for some function $\hat{p} : \mathcal{A}^* \to \Delta$ possibly based on $\mathcal{H}$ but not necessarily in $\mathcal{H}$, where $\omega_{1:t-1}$ denotes the string $\omega_1 \ldots \omega_{t-1}$.

Regret

Our goal is to minimize the *regret*.

Definition. (Regret) For the individual sequence protocol, the *regret* is given by

$$R_T(h) \coloneqq \max_{s \in \mathcal{A}^T} \left[\sum_{t=1}^{T} \lambda(s_t, \hat{p}(s_{1:t-1})) - \sum_{t=1}^{T} \lambda(s_t, h(s_{1:t-1})) \right], \quad \forall h \in \mathcal{H}.$$

For convenience, we define

$$R_T(\mathcal{H}) \coloneqq \max_{h \in \mathcal{H}} R_T(h).$$

N. Merhav and M. Feder. 1998. Universal prediction.

Regret bound

Theorem 2. Suppose the hypothesis class $\mathcal{H}$ is countable. Let $\pi(h)$ be a probability distribution on $\mathcal{H}$. Then, there exists an algorithm that yields

$$R_T(h) \leq \log\left(\frac{1}{\pi(h)}\right), \quad \forall h \in \mathcal{H}.$$

Corollary 1. Suppose the hypothesis class $\mathcal{H}$ is finite. Then, there exits an algorithm that yields

$$R_T \leq \log|\mathcal{H}|.$$

A. DeSantis *et al.* 1988. Learning probabilistic prediction functions.

Mixture forecaster

Define

$$\tilde{h}(\omega_{1:t}) \coloneqq \prod_{\tau=1}^{t} h(\omega_\tau|\omega_{1:\tau-1}), \quad \forall t \in \mathbb{N}, \omega_{1:t} \in \mathcal{A}^t,$$

where $h(\omega_\tau|\omega_{1:\tau-1})$ denotes the ω_τ-th entry of the vector $h(\omega_{1:\tau-1})$.

Algorithm. (Mixture forecaster) For every $t \in \mathbb{N}$, set

$$\hat{p}(a|\omega_{1:t-1}) \leftarrow \frac{\sum_{h\in\mathcal{H}} \pi(h) h(a|\omega_{1:t-1})\tilde{h}(\omega_{1:t-1})}{\sum_{h\in\mathcal{H}} \pi(h)\tilde{h}(\omega_{1:t-1})}, \quad \forall a \in \mathcal{A},$$

where $\hat{p}(a|\omega_{1:t-1})$ denotes the a-th entry of the vector $\hat{p}(\omega_{1:t-1})$.

Probabilistic interpretation of the mixture forecaster (1/2)

$$\hat{p}(a|\omega_{1:t-1}) \leftarrow \frac{\sum_{h\in\mathcal{H}} \pi(h)h(a|\omega_{1:t-1})\tilde{h}(\omega_{1:t-1})}{\sum_{h\in\mathcal{H}} \pi(h)\tilde{h}(\omega_{1:t-1})}, \quad \forall a \in \mathcal{A}.$$

- View $h(a|\omega_{1:t-1})$ as a *conditional probability* of the event $\{\, \omega_t = a \,\}$ given $\omega_{1:t-1}$.
- View $\{\, \pi(h) \mid h \in \mathcal{H} \,\}$ as a *prior distribution* on $\mathcal{H}$.
- Suppose that the string $\omega_1\omega_2\ldots$ is generated by a stochastic process $\{\, s_t \mid t \in \mathbb{N} \,\}$ following

$$\mathsf{P}\left(s_{1:t} = \omega_{1:t}\right) = \sum_{h\in\mathcal{H}} \pi(h) \prod_{\tau=1}^{t} h(\omega_\tau|\omega_{1:\tau-1}).$$

Probabilistic interpretation of the mixture forecaster (2/2)

Proposition 3. Under the assumptions in the previous slide, it holds that

$$\hat{p}(a|\omega_{1:t-1}) = \mathsf{P}\left(s_t = a | s_{1:t-1} = \omega_{1:t-1}\right), \quad \forall a \in \mathcal{A}.$$

Proof. We write

$$\begin{aligned} \hat{p}(a|\omega_{1:t-1}) &= \frac{\sum_{h\in\mathcal{H}} \pi(h) h(a|\omega_{1:t-1}) \tilde{h}(\omega_{1:t-1})}{\sum_{h\in\mathcal{H}} \pi(h) \tilde{h}(\omega_{1:t-1})} \\ &= \frac{\mathsf{P}\left(s_t = a, s_{1:t-1} = \omega_{1:t-1}\right)}{\mathsf{P}\left(s_{1:t-1} = \omega_{1:t-1}\right)} \\ &= \mathsf{P}\left(s_t = a | s_{1:t-1} = \omega_{1:t-1}\right). \end{aligned}$$

The last equality follows from *Baye's theorem*.

Proof of Theorem 2

Proof. (Theorem 2) We write

$$\begin{aligned}
\sum_{t=1}^{T} \lambda(\omega_t, \gamma_t) &= \sum_{t=1}^{T} -\log \hat{p}(\omega_t|\omega_{1:t-1}) \\
&= \sum_{t=1}^{T} -\log \left[\frac{\sum_{h\in\mathcal{H}} \pi(h) h(\omega_t|\omega_{1:t-1}) \tilde{h}(\omega_{1:t-1})}{\sum_{h\in\mathcal{H}} \pi(h) \tilde{h}(\omega_{1:t-1})} \right] \\
&= -\log \left\{ \prod_{t=1}^{T} \left[\frac{\sum_{h\in\mathcal{H}} \pi(h) h(\omega_t|\omega_{1:t-1}) \tilde{h}(\omega_{1:t-1})}{\sum_{h\in\mathcal{H}} \pi(h) \tilde{h}(\omega_{1:t-1})} \right] \right\} \\
&= -\log \left[\frac{\sum_{h\in\mathcal{H}} \pi(h) h(\omega_T|\omega_{1:T-1}) \tilde{h}(\omega_{1:T-1})}{\sum_{h\in\mathcal{H}} \pi(h)} \right] \\
&\leq -\log \left[\pi(h) \prod_{t=1}^{T} h(\omega_t|\omega_{1:t-1}) \right], \quad \forall h \in \mathcal{H}.
\end{aligned}$$

Application: Universal source coding

Data compression problem

Let $\mathcal{A} = \{ a_1, \ldots, a_m \}$ be an alphabet. Given a string $\omega_{1:T} \in \mathcal{A}^T$, how do we compress it using as few bits as possible?

Claude Shannon's idea. Suppose that the string is generated following a probability distribution. Consider the *expected number of bits*

C. E. Shannon. 1948. A mathematical theory of communication.

Source codes

Let $\mathcal{B} = \{0, 1\}$.

Definition. (Source code) A *source code* is a mapping from $\mathcal{A}$ to the set of *codewords* (a subset of $\mathcal{B}^*$).

Definition. (Extension) The *extension* of a source code maps a string in $\mathcal{A}^*$ to a sequence of codewords.

Definition. (Uniquely decodable codes) A source code is *uniquely decodable*, if its extension is injective.

Definition. (Prefix code) A source code is a *prefix code*, if no codeword is the prefix of another codeword.

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Kraft's inequality

Let $\ell(a_i)$ be the length of the codeword associated with $a_i \in \mathcal{A}$.

Theorem 3. (Kraft's inequality) A prefix code must satisfy

$$\sum_{a_i \in \mathcal{A}} 2^{-\ell(a_i)} \leq 1.$$

Moreover, for any ℓ satisfying the inequality, there exists a corresponding prefix code.

Theorem 4. (McMillan's inequality) The same statements also hold for uniquely decodable codes.

L. G. Kraft. 1949. A device for quantizing, grouping, and coding amplitude-modulated pulses.

B. McMillan. 1956. Two inequalities implied by unique decipherability.

Implications of inequalities of Kraft and McMillan

Definition. (Complete code) A uniquely decodable code satisfying Kraft's inequality with equality is called a *complete code*.

Remark. If a uniquely decodable code is not complete, then it has some redundancy.

Observation. *The length function ℓ of a complete code defines a probability distribution* (up to the rounding error), with

$$p_i := 2^{-\ell(a_i)}, \quad 1 \le i \le m.$$

Observation. It suffices to consider complete prefix codes.

Source coding theorem

Definition. (Shannon entropy) The *Shannon entropy* (of base 2) of a random variable ξ taking values in a finite set $\mathcal{X}$ is given by

$$H(\xi) := \sum_{x \in \mathcal{X}} \mathsf{P}\left(\xi = x\right) \log_2 \frac{1}{\mathsf{P}\left(\xi = x\right)}.$$

Theorem 4. (Source coding theorem) Denote by $L_T^\star$ the minimum expected number of bits per symbol. There is a uniquely decodable source code that achieves

$$H(\omega_{1:T}) \leq T L_T^\star \leq H(\omega_{1:T}) + 1.$$

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Proof of Theorem 4 (1/2)

Sketch of proof. (Theorem 4) View $\omega_{1:T}$ as a random element in $\mathcal{A}^T$. Consider complete prefix codes $\mathcal{C} : \mathcal{A}^T \to \{0, 1\}^*$.

Recall the equivalence between codeword lengths and probability distributions for a complete prefix code. Then, we have

$$TL_T^\star = \min_q \sum_{a_{1:T} \in \mathcal{A}^T} p(a_{1:T}) \log_2 \frac{1}{q(a_{1:T})},$$

subject to the constraint that q defines a probability distribution on $\mathcal{A}^T$, where

$$p(a_{1:T}) := \mathsf{P}\left(\omega_{1:T} = a_{1:T}\right), \quad \forall a_{1:T} \in \mathcal{A}^T.$$

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Proof of Theorem 4 (2/2)

Sketch of proof continued. (Theorem 4) Define

$$f(q) \coloneqq \sum_{a_{1:T} \in \mathcal{A}^T} p(a_{1:T}) \log_2 \frac{1}{q(a_{1:T})}.$$

Then, we have

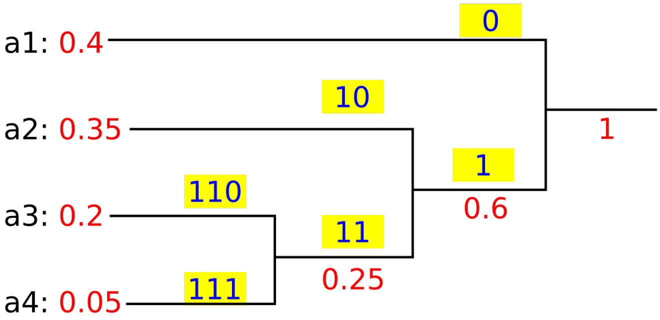
$$f(q) - f(p) = \sum_{a_{1:T} \in \mathcal{A}^T} p(a_{1:T}) \log_2 \frac{p(a_{1:T})}{q(a_{1:T})} = \frac{D(p\|q)}{\log 2}.$$

It remains to notice that $D(p\|q) \geq 0$ (known as *Gibbs' inequality*), and apply Kraft's inequality.

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Huffman code

Theorem 5. Theorem 4 is achieved by the *Huffman code*.



"Huffman coding" in *Wikipedia*.

Universal source coding (1/2)

Suppose we do not have access to the exact probability distribution p, but we know that $p \in \mathcal{P}$ for a class of probability distributions $\mathcal{P}$. How do we compress any given string *almost optimally*?

By Bayes' rule, each probability distribution $q \in \mathcal{P}$ defines a *conditional distribution function* $h : \mathcal{A}^* \to \Delta$, as

$$h(a_t | a_{1:t-1}) := \frac{q(a_{1:t})}{q(a_{1:t-1})}, \quad \forall t \in \mathbb{N}, a_{1:t} \in \mathcal{A}^*.$$

Then, a class of probability distributions defines a *hypothesis class* $\mathcal{H}$ of conditional distribution functions.

Universal source coding (2/2)

Notice that

$$
\begin{aligned}
-\log \mathsf{P}\left(\omega_{1:T} = a_{1:T}\right) &= -\log \prod_{t=1}^{T} \mathsf{P}\left(\omega_t = a_t | \omega_{1:t-1} = a_{1:t-1}\right) \\
&= \sum_{t=1}^{T} \left[-\log \mathsf{P}\left(\omega_t = a_t | \omega_{1:t-1} = a_{1:t-1}\right)\right].
\end{aligned}
$$

Our goal is then to output probability distributions $\gamma_1, \ldots, \gamma_T$, such that

$$
\sum_{t=1}^{T} -\log(\gamma_t(\omega_t)) \approx \min_{h \in \mathcal{H}} \sum_{t=1}^{T} -\log h(\omega_t | \omega_{1:t-1}).
$$

That is, we arrive at the individual sequence prediction problem introduced before.

Shannon-Fano-Elias code (1/3)

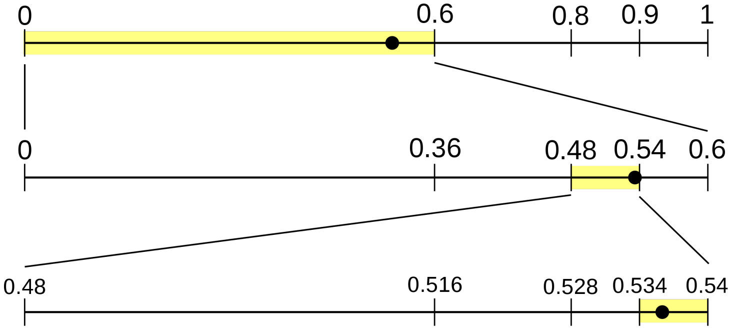
The *Shannon-Fano-Elias code* allows efficient sequential encoding and decoding procedures.

Theorem. The expected number of bits per symbol $\bar{L}_n$ of the *Shannon-Fano-Elias code* satisfies

$$T\bar{L}_n \leq H(\omega_1, \ldots, \omega_T) + 2.$$

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Shannon-Fano-Elias code (2/3)



"Arithmetic coding" in *Wikipedia*.

Shannon-Fano-Elias code (3/3)

Remaining steps in encoding.

1. Choose the middle point in the interval associated with $\omega_{1:T}$.
2. Represent the middle point into bits $0.b_1 b_2 \dots$
3. Round the number to $0.b_1 b_2 \dots b_{\ell(\omega_{1:T})}$, where

$$\ell(\omega_{1:T}) \coloneqq \lceil -\log \mathsf{P}(s_{1:T} = \omega_{1:T}) \rceil + 1.$$

4. Output $b_1 b_2 \dots b_{\ell(\omega_{1:T})}$ as the compressed data.

Decoding. Check the associated interval.

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*.

Conclusions

Conclusions

- We have introduced the notions of competitiveness and regret.
- Weighted majority vote is 2.41-competitive in the voting protocol.
- Mixture forecasting achieves a $\log|\mathcal{H}|$ regret in the finite hypothesis class setting, in the individual sequence prediction protocol.
- Mixture forecasting has a probability interpretation.
- A motivation for the individual sequence prediction problem is universal coding.

Common pattern (1/2)

Weighted majority vote. (Equivalent formulation) Denote the weight for $\textsc{Expert}$ i after the t-th round by $\pi_{t+1}(i)$.

- Set $\{\, \pi_0(i) \mid 1 \le i \le m \,\}$ to be the uniform probability distribution on $\{\, 1, \dots, m \,\}$.
- For every $t \in \{\, 0 \,\} \cup \mathbb{N}$, output

$$\gamma_t = \begin{cases} 0 & \text{, if } \sum_{i:\gamma_t(i)=0} \pi_t(i) \ge 1/2, \\ 1 & \text{, otherwise,} \end{cases}$$

  and set

$$\pi_{t+1}(i) = \frac{\pi_t(i) \mathrm{e}^{-\eta\lambda(\omega_t, \gamma_t(i))}}{\sum_{1 \le i \le m} \pi_t(i) \mathrm{e}^{-\eta\lambda(\omega_t, \gamma_t(i))}}, \quad \forall 1 \le i \le m,$$

  where $\eta = \log(1/2)$.

Common pattern (2/2)

Mixture forecaster. (Equivalent formulation) Denote the weight for a hypothesis $h \in \mathcal{H}$ after the t-th round by $\pi_{t+1}(h)$.

- Set $\{ \pi_0(h) \mid h \in \mathcal{H} \}$ to be any probability distribution on $\mathcal{H}$.
- For every $t \in \{ 0 \} \cup \mathbb{N}$, output

  $$\gamma_t = \sum_{h \in \mathcal{H}} h(\omega_{1:t-1}) \pi_t(h) \in \Delta,$$

  and set

  $$\pi_{t+1}(h) = \frac{\pi_t(h) \mathrm{e}^{-\eta \lambda(\omega_t, h(\omega_{1:t-1}))}}{\sum_{h \in \mathcal{H}} \pi_t(h) \mathrm{e}^{-\eta \lambda(\omega_t, h(\omega_{1:t-1}))}}, \quad \forall h \in \mathcal{H},$$

  where $\eta = 1$.

Next lecture

- Individual sequence prediction continued.