(F1X) Teoria dell'Informazione e della Trasmissione

15 – Informazione e Predizione

Docente: *Nicolò Cesa-Bianchi* versione 14 maggio 2013

Consideriamo una sorgente $\langle \mathcal{X}, p \rangle$. Dallo studio della codifica sorgente, sappiamo che la lunghezza media delle parole del miglior codice $c : \mathcal{X} \to \{0,1\}^*$ univocamente decodificabile è compresa fra $H(X)$ e $H(X)+1$ e che questo codice ottimo è anche istantaneo.

Diamo ora un significato diverso a $H(X)$. Supponiamo di dover indovinare il valore di un simbolo sorgente x estratto dalla distribuzione p tramite una serie di domande binarie (sì/no) che possiamo scegliere liberamente. È facile vedere che posso usare l'albero di un codice istantaneo ottimo c^* per formulare le domande. Ricordiamo che questo albero è binario e ha gli elementi di $\mathcal{X}$ sulle foglie. Il cammino dalla radice a una foglia x' identifica la parola di codice $c^*(x')$ ed ha lunghezza $\ell_{c^*}(x)$.

L'algoritmo per formulare le domande inizialmente pone la radice come nodo corrente. La prima domanda che facciamo è se x appartiene alle foglie del sottoalbero sinistro della radice. Se la risposta e sì, il figlio sinistro della radice diventa il nodo corrente, altrimenti il figlio destro della radice diventa il nodo corrente. Quindi procediamo come prima, chiedendo se x appartiene alle foglie del sottoalbero sinistro della radice. Procedendo in questo modo, dopo $\ell_{c^*}(x)$ domande abbiamo identificato x. Perciò, in media facciamo un numero di domande pari a $\mathbb{E}\big[\ell_{c^*}(X)\big]$, che per un codice ottimo è compreso fra $H(X)$ e $H(X)+1$.

Supponiamo ora che x non sia estratta da p ma sia un elemento arbitrario di $\mathcal{X}$. In questo caso posso formulare le domande usando un codice istantaneo ottimo c^* per la sorgente dove p è uniforme, ovvero $p(x) = \frac{1}{m}$ per ogni $x \in \mathcal{X}$ dove $m = |\mathcal{X}|$. Le lunghezze delle parole di questo codice avranno due soli possibili valori: $\lfloor \log_2 m \rfloor$ e $\lceil \log_2 m \rceil$ (che coincidono quando m è potenza di due). Quindi, applicando l'algoritmo precedente per formulare le domande, posso identificare *ogni* x usando al più $\lceil \log_2 m \rceil$ domande. Si noti che questo valore è nuovamente compreso fra $H(X)$ e $H(X)+1$ dove $H(X) = \log_2 m$ è l'entropia della distribuzione uniforme su $\mathcal{X}$.

Consideriamo ora un problema diverso, ma in qualche modo collegato: quello della predizione sequenziale tramite aggregazione di esperti. Prendiamo come esempio il problema delle previsioni del tempo. Supponiamo di dover predire, ogni giorno e per un anno di fila, se il giorno successivo pioverà o meno in una data località fissata. In astratto, dobbiamo quindi predire una sequenza ignota $(y_1, y_2, \dots, y_T)$, dove $y_t = 0$ se il t-esimo giorno non piove e $y_t = 1$ se il t-esimo giorno piove, con $T = 365$. Denotiamo con $\widehat{y}_t$ la nostra previsione di y_t.

Fissiamo un dato insieme di N esperti per le previsioni del tempo (per esempio pacchetti software o metereologi in carne ed ossa). Le predizioni si svolgono nel modo seguente: per ogni giorno $t = 1, 2, \dots$

1. al termine del giorno precedente raccolgo le previsioni $x_{1,t}, \dots, x_{N,t} \in \{0,1\}$ degli esperti per il giorno corrente e formulo la previsione $\widehat{y}_t \in \{0,1\}$
2. al termine del giorno corrente osservo se ha piovuto o meno e ricavo il bit corretto $y_t \in \{0,1\}$

3. se $\widehat{y}_t \neq y_t$, ho commesso un errore, analogamente se $x_{i,t} \neq y_t$ l'esperto i ha commesso un errore.

Alla fine dell'anno ci sarà un esperto che ha sbagliato meno degli altri (il miglior esperto dell'anno). Il nostro primo obiettivo è quello di sviluppare un algoritmo che, qualsiasi cosa succeda, non sbaglia molto di più di questo miglior esperto. Indichiamo con

$$M_{i,t} = \sum_{s=1}^{t} \mathbb{I}\{x_{i,s} \neq y_s\}$$

il numero di errori dell'esperto i nei primi t passi di predizione. Analogamente, indichiamo con

$$M_t = \sum_{s=1}^{t} \mathbb{I}\{\widehat{y}_s \neq y_s\}$$

il numero di errori compiuti dall'algoritmo di predizione nello stesso intervallo di tempo.

Per iniziare, assumiamo che ci sia un esperto che non fa neanche un errore di previsione durante tutto l'anno, ovvero

$$\min_{i=1,\ldots,N} M_{i,T} = 0 \ .$$

Qual è una buona strategia di previsione in questo caso?

Consideriamo la seguente strategia detta Halving: ogni giorno, predici come la maggioranza degli esperti fra tutti quelli che non hanno mai sbagliato in precedenza (se esattamente la metà degli esperti fra quelli che non hanno mai sbagliato predice 0, allora predici 0). Non è difficile calcolare un maggiorante sul numero di sbagli di Halving. Sia S_t il sottoinsieme di esperti che hanno azzeccato le prime t predizioni, ovvero

$$S_t = \{1 \leq i \leq N \ : \ M_{i,t} = 0\} \ .$$

Quindi Halving predice $\widehat{y}_{t+1} = 1$ sse

$$|\{i \in S_t \ : \ x_{t+1,i} = 1\}| \geq |\{j \in S_t \ : \ x_{t+1,j} = 0\}|$$

Abbiamo:

1. $S_0 = \{1, \ldots, N\}$.
2. $|S_T| \geq 1$, dato che almeno un esperto non sbaglia mai.
3. $|S_{t+1}| \leq |S_t|$ per ogni t.
4. $|S_{t+1}| \leq |S_t|/2$ per ogni t tale che $\widehat{y}_t \neq y_t$.

L'ultima disuguaglianza è dovuta al fatto che, se sbagliamo predicendo con la maggioranza di quelli che non avevano mai sbagliato prima, allora hanno sbagliato almeno metà di quelli che non avevano mai sbagliato prima.

Ricordiamo che M_T è numero (incognito) di sbagli di Halving nei T passi di predizione. Allora

$$1 \leq |S_T| \leq \frac{|S_0|}{2^{M_T}} = \frac{N}{2^{M_T}} \ .$$

Risolvendo per M_T otteniamo $M_T \leq \log_2 N$.

Dato che M_T deve essere intero, otteniamo infine $M_T \leq \lfloor \log_2 N \rfloor$.

Per esempio, con 1000 esperti faremo al più 9 sbagli. Inoltre, raddoppiando il numero di esperti, facciamo al massimo uno sbaglio in più, infatti

$$\lfloor \log_2(2N) \rfloor = \lfloor \log_2 N + 1 \rfloor = \lfloor \log_2 N \rfloor + 1 \ .$$

Si osservi che:

1. Questo algoritmo può essere utilizzato per qualsiasi problema di predizione binaria.
2. Il maggiorante che abbiamo dimostrato dipende solo dal numero degli esperti, ma non dalle loro predizioni, né dalla sequenza binaria $y_1, \ldots, y_T$, né dalla sua lunghezza T (ovvero, con 1000 esperti facciamo al più 9 errori anche su sequenze infinite).

Una formulazione equivalente di Halving è la seguente: a ogni esperto i è associato al tempo t un peso positivo $w_{i,t}$. All'inizio $w_{i,1} = 1/N$ per ogni esperto i. Se i sbaglia al tempo t, allora il suo peso viene moltiplicato per 0, cioè $w_{i,t+1} = 0$. Per predire, confrontiamo la somma dei pesi degli esperti che predicono 1 con la somma dei pesi degli esperti che predicono 0:

$$\widehat{y}_t = 1 \quad \text{se e solo e} \quad \sum_{i \,:\, x_{t,i}=1} w_{i,t} \geq \sum_{j \,:\, x_{t,j}=0} w_{j,t} \ .$$

L'analisi è molto simile alla precedente con W_t che svolge il ruolo di $|S_{t-1}|$. Sia $W_t = w_{1,t} + \cdots + w_{N,t}$. Infatti,

$$\frac{1}{N} \leq W_{T+1} \leq \frac{W_1}{2^{M_T}} = \frac{1}{2^{M_T}} \ .$$

Risolvendo per M_T otteniamo $M_T \leq \lfloor \log_2 N \rfloor$.

Se invece $w_{i,1} = p_i > 0$ per ogni $i = 1, \ldots, N$ dove $p_1 + \cdots + p_N = 1$, ripetendo l'analisi di Halving otteniamo

$$M_T \leq \log_2 \frac{1}{p_k}$$

doke k è l'esperto tale che $M_{k,T} = 0$. Nel caso in cui l'esperto ottimo k sia estratto proprio dalla distribuzione $(p_1, \ldots, p_N)$ usata per assegnare i pesi iniziali agli esperti, otteniamo che il numero medio di errori rispetto all'estrazione di k è

$$\mathbb{E}\big[M_T\big] \leq \sum_{k=1}^{N} p_k \log_2 \frac{1}{p_k} = H(X)$$

ovvero il numero medio di errori è l'entropia della distribuzione $(p_1, \ldots, p_N)$.

Occupiamoci ora del caso (molto più verosimile) in cui non ci sia un esperto che fa sempre previsioni esatte. Vediamo se è possibile adattare Halving a questo caso. L'algoritmo può essere riformulato nel modo seguente.

Per renderla più morbida, decidiamo che ad ogni sbaglio, il peso di un esperto debba essere moltiplicato per $0 \leq \beta < 1$, cioè $w_i \leftarrow \beta \times w_i$. Per $\beta = 0$, otteniamo nuovamente Halving. Per $0 < \beta < 1$, otteniamo un algoritmo diverso (in realtà, otteniamo un algoritmo diverso per ogni valore di β nell'intervallo considerato). Chiamiamo questa famiglia di algoritmi Weighted Majority (WM). L'analisi è simile a quella di Halving.

Sia W_{t+1} il peso totale degli esperti dopo t predizioni. Abbiamo:

1. $W_{t+1} \leq W_t$ per ogni t.
2. $W_{t+1} \leq (1/2 + \beta/2)W_t$ per ogni t tale che $\widehat{y}_t \neq y_t$.

L'ultima disuguaglianza è dovuta al fatto che, se sbagliamo predicendo con la maggioranza pesata, allora il peso degli esperti che sbagliano è almeno la metà del peso corrente. Quindi, almeno la metà del peso corrente viene moltiplicata per β.

Denotiamo con M_T il numero (incognito) di sbagli di WM dopo T predizioni. Allora, usando il fatto che $W_1 = N$,

$$W_T \leq \left(\frac{1}{2} + \frac{\beta}{2}\right)^{M_T} W_1 = \left(\frac{1+\beta}{2}\right)^{M_T} N \ .$$

Risolvendo per M_t otteniamo

$$M_T \leq \left\lfloor \frac{\log_2(N/W_{T+1})}{\log_2\left(\frac{2}{1+\beta}\right)} \right\rfloor \ .$$

La quantità W_{T+1} denota la somma dei pesi degli esperti dopo T predizioni.

Ora, il peso di ciascun esperto i parte da 1 e viene moltiplicato per β ogni volta che l'esperto sbaglia. Perciò, dopo T predizioni avremo $w_{i,T+1} = \beta^{M_{i,T}}$, dove $M_{i,T} = \sum_{t=1}^{T} \mathbb{I}\{x_{t,i} \neq y_t\}$ è il numero totale di sbagli dell'esperto i.

Quindi, posto $M^* = \min_{i=1,\ldots,N} M_{i,T}$ il numero totale di sbagli del miglior esperto, abbiamo

$$W_{T+1} = \sum_{i=1}^{N} \beta^{M_{i,T}} \geq \max_{i=1,\ldots,N} \beta^{M_{i,T}} = \beta^{M^*} \ .$$

Sostituendo nel maggiorante per M_T otteniamo infine

$$M_T \leq \frac{\log_2 N + M^* \log_2(1/\beta)}{\log_2\left(\frac{2}{1+\beta}\right)} \ . \tag{1}$$

Per esempio, se $\beta = 1/2$ allora

$$M_T \leq \frac{\log_2 N + M^* \log_2(1/\beta)}{\log_2\left(\frac{2}{1+\beta}\right)} = \frac{\log_2 N + M^*}{\log_2(4/3)} < 2.41(\log_2 N + M^*) \ .$$

Quindi, se il migliore fra 10 esperti sbaglia 100 volte, allora WM con $\beta = 1/2$ sbaglierà al più 248 volte.

Occupiamoci ora della scelta del parametero β. Definiamo la funzione

$$f(\beta) = \frac{\log_2 N + M^* \log_2(1/\beta)}{\log_2\left(\frac{2}{1+\beta}\right)}$$

per $0 \leq \beta \leq 1$. Studiando la derivata di f troviamo che $f(\beta)$ è minimizzata per $\beta' = M^*/(M'-M^*)$, dove M' è l'unica soluzione (per $M > 2M^*$) dell'equazione

$$M = \log_2 N + M \times H(M^*/M)$$

e H è la funzione di entropia binaria $H(x) = -x\log_2(x) - (1-x)\log_2(1-x)$. Sostituendo, scopriamo anche che $f(\beta') = M'$. Ora, usando la maggiorazione $H(q) \leq 2\sqrt{q(1-q)}$, possiamo scrivere

$$M' = \log_2 N + M' \times H(M^*/M') \leq \log_2 N + M'\sqrt{\frac{M^*}{M'}\left(1 - \frac{M^*}{M'}\right)} \leq \log_2 N + \sqrt{M^*(M'-M^*)}$$

Sostituendo β_1 nel membro destro di (1) otteniamo il nuovo maggiorante

$$M \leq 2M^* + \log_2 N + 2\sqrt{M^* \log_2 N} \tag{2}$$

dove, come prima, M indica il numero di sbagli compiuti da WM su una sequenza arbitraria e M^* indica il numero di sbagli compiuti dal miglior esperto sulla stessa sequenza.

Si noti che in (2) il fattore moltiplicativo di M^* (che possiamo considerare il termine dominante in quanto ci aspettiamo che cresca al crescere del numero di predizioni) è 2, mentre in (1) lo stesso fattore è 2.41 .

Però c'è un problema: la scelta di β che dà (2) dipende dal numero N di esperti (che è noto) e dal numero M^* di sbagli del miglior esperto, che conosceremo solo alla fine!

D'altra parte, esaminando (1) notiamo che l'unica scelta di β costante (cioè indipendente da M^*) che ci dà un fattore 2 di fronte a M^* è $\beta = 1$. Ma questa scelta non va bene in quanto il fattore moltiplicativo di $\log_2 N$ nella stessa espressione diventa infinito.

Se supponiamo di conoscere il numero T di predizioni che vogliamo fare, possiamo ottenere qualcosa di simile a (2). Infatti, è possibile dimostrare che la scelta

$$\beta_2 = g\left(\sqrt{\ln(N)/T}\right) \qquad \text{dove} \qquad g(x) = \frac{1}{1 + 2x + x^2/\ln(2)}$$

nel membro destro di (1) produce il maggiorante

$$M \leq \left\lfloor 2M^* + \log_2 N + 2\sqrt{T \ln N} \right\rfloor \ .$$