Mistake Bounds per algoritmi On-line

Quando si considerano algoritmi on-line per l'apprendimento di concetti, è ragionevole essere interessati ad un limite superiore al numero di errori commessi prima di apprendere *esattamente* il concetto target

Il **Modello Mistake Bound** è stato definito per questo scopo:

- le istanze x_i sono presentate ad L una alla volta
- data una istanza x, L deve "indovinare" il valore target $c(x)$
- solo dopo, il valore corretto è fornito ad L ai fini dell'apprendimento
- se la predizione di L era sbagliata, allora si ha un errore (mistake)

Bisogna rispondere alla seguente domanda:

> "Quanti errori farà L prima di apprendere esattamente il concetto target ?"

Mistake bound per la versione on-line di Find-S

L'algoritmo **Find-S** può essere usato in versione on-line !

/* versione on-line di Find-S per congiunzione di m letterali */

inizializza h alla ipotesi più specifica $h \equiv l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \wedge \cdots \wedge l_m \wedge \neg l_m$

do forever /* in effetti, fino a quando arrivano esempi */

leggi la nuova istanza x e predici $h(x)$

leggi $c(x)$, cioè il valore target per x

/* errore ! */
if $(h(x) \neq c(x))$ rimuovi da h ogni letterale che non è soddisfatto da x

Assumendo $c \in \mathcal{H}$, e assenza di rumore negli esempi

E' possibile dare un limite superiore al numero di errori ?

Mistake bound per la versione on-line di Find-S

E’ possibile dare un limite superiore al numero di errori ?

Si!

- l’ipotesi iniziale contiene $2m$ letterali
- dopo il primo sbaglio (che occorre subito!), solo m letterali rimangono nella ipotesi corrente
- dopo ogni altro errore, almeno un letterale è rimosso dalla ipotesi corrente

Quindi il numero totale di errori che **Find-S** commette prima di convergere alla ipotesi corretta è $\leq m + 1$

Mistake bound per Halving

L'algoritmo **Halving** è una versione on-line di **Candidate-Elimination**

/* Aggiorna il Version Space (VS) on-line */

inizializza S e G come in **Candidate-Elimination**

do forever /* in effetti, fino a che VS contiene più di 1 ipotesi */

leggi una nuova istanza x e predici tramite voto a maggioranza delle ipotesi in VS

leggi $c(x)$, cioè il valore target per x

aggiorna S e G in modo da rimuovere da VS le ipotesi h per cui $h(x) \neq c(x)$

/* un errore occorre solo se la maggioranza delle ipotesi in VS sono sbagliate */

Assumendo $c \in \mathcal{H}$, $|\mathcal{H}| < \infty$, e nessun rumore negli esempi

E' possibile dare un limite superiore al numero di errori ?

Mistake bound per Halving

E' possibile dare un limite superiore al numero di errori ?

Si!

- il VS iniziale contiene $|\mathcal{H}|$ ipotesi
- dopo un errore (che occorre quando il voto a maggioranza è sbagliato), ALMENO $|VS|/2$ ipotesi sono rimosse da VS

Quindi il numero totale di errori che **Halving** commette prima di convergere alla ipotesi corretta è $\leq log_2|\mathcal{H}|$

ATTENZIONE: **Halving** può convergere alla ipotesi corretta senza errori !! Accade quando il voto a maggioranza è sempre corretto e solo le ipotesi minoritarie sono rimosse da VS

Optimal Mistake Bounds

Fino ad ora abbiamo considerato limiti al numero di errori (caso pessimo) per algoritmi *specifici*.

E' possibile dare il più basso fra i limiti di errore su tutti i possibili algoritmi di apprendimento (*optimal mistake bound*) ?

- Assumiamo $C = \mathcal{H}$
- Sia $M_L(c)$ il massimo su tutte le possibili sequenze di esempi di apprendimento del numero di errori commesso da L per apprendere esattamente il concetto $c \in C$
- Sia $M_L(C) \equiv \max_{c \in C} M_L(c)$

Definizione: Sia C una classe non vuota di concetti. L' **optimal mistake bound** per C, denotato *Opt(C)*, è il minimo su tutti i possibili algoritmi di apprendimento L di $M_L(C)$:

$$Opt(C) \equiv \min_{L \in AlgoritmiApprendimento} M_L(C)$$

Optimal Mistake Bounds

Littlestone (1987) ha mostrato che

$$\boxed{VC(C) \leq Opt(C) \leq M_{Halving}(C) \leq log_2(|C|)}$$

Un esempio finale di Mistake Bound...

Fino ad ora abbiamo visto esempi di algoritmi di apprendimento capaci di trattare con esempi di apprendimento consistenti (cioè che non danno luogo a contraddizioni)

Non è difficile modificare **Halving** in modo da ottenere un algoritmo capace di trattare dati inconsistenti: Algoritmo **Weighted-Majority**

- l'idea base è di assegnare ad ogni ipotesi (ma possiamo estendere l'idea ad ogni insieme di predittori) un peso che è usato per pesare il voto associato ad ogni ipotesi
- quindi, invece di considerare il voto a maggioranza, consideriamo il voto a maggioranza pesata
- quando una ipotesi commette un errore, invece di rimuoverla, si moltiplica il peso a lei associata per un fattore $\beta < 1$ (riduzione del peso)

Weighted-Majority

/* Weighted-Majority: l' algoritmo Halving... con pesi! */

per ogni ipotesi (o predittore) h_j definire il peso w_j, inizialmente uguale a 1

do forever /* in effetti, fino a che ci sono esempi */

leggi una nuova istanza x

calcola $q_0 \leftarrow \sum_{j:h_j(x)=0} w_j$ e $q_1 \leftarrow \sum_{j:h_j(x)=1} w_j$

if $q_0 > q_1$ **then** predici 0

if $q_1 > q_0$ **then** predici 1

if $q_0 = q_1$ **then** predici 0 o 1 a caso

read $c(x)$, i.e. the target value for x

for each h_j **do**

if $h_j(x) \neq c(x)$ **then** $w_j \leftarrow \beta w_j$ /* $0 \leq \beta < 1$ */

Cosa accade se $\beta = 0$?

Mistake bound per Weighted-Majority

Teorema: Sia $seq \equiv (x_1, c(x_1)), (x_2, c(x_2)), \ldots$ una qualunque sequenza di esempi di allenamento e sia k il numero minimo di errori commesso su seq da una qualunque ipotesi dello Spazio delle Ipotesi. Allora il numero di errori su seq commesso da **Weighted-Majority** usando $\beta = \frac{1}{2}$ è al più

$$\boxed{2.4(k + \log_2(|\mathcal{H}|))}$$

(lo proviamo!)

Per $0 \leq \beta < 1$, Littlestone e Warmuth (1991) hanno provato che il bound di sopra diventa

$$\boxed{\frac{k \log_2 \frac{1}{\beta} + \log_2(|\mathcal{H}|)}{\log_2 \frac{2}{1+\beta}}}$$

(questo non lo proviamo...)

Prova...

Per provare il teorema confrontiamo il peso finale w^* della migliore ipotesi h^*, cioè quella che produce il numero minore di errori k, con la somma finale dei pesi su tutte le ipotesi $W = \sum_{j=1}^{|\mathcal{H}|} w_j$ (naturalmente abbiamo $w^* \leq W$)

- **fatto 1:** $w^* = (\frac{1}{2})^k$, infatti h^* commette esattamente k errori
- **fatto 2:** per ogni errore di **Weighted-Majority**, W si riduce di al più $\frac{3}{4}W$, infatti se **Weighted-Majority** commette un errore, tutte le ipotesi h_j che hanno contribuito alla maggioranza pesata avranno il loro peso ridotto di metà; quindi almeno $\frac{1}{2}W$ (il voto a maggioranza pesata è $\geq \frac{1}{2}W$) del peso totale è ridotto a metà, cioè il nuovo peso totale è **minore o uguale a**

$$\underbrace{\frac{1}{2}W}_{\text{minoranza}} + \underbrace{\frac{1}{2}(\frac{1}{2}W)}_{\text{maggioranza}} = \frac{3}{4}W$$

- **fatto 3:** se M è il numero totale di errori prodotti da **Weighted-Majority**, allora per il peso totale finale $W \leq |\mathcal{H}|(\frac{3}{4})^M$, a causa del fatto 2 e la condizione iniziale $W = |\mathcal{H}|$

Prova...

Ricordando che $w^* \leq W$, ed a causa dei fatti 1-3, abbiamo

$$(\frac{1}{2})^k = w^* \leq W \leq |\mathcal{H}|(\frac{3}{4})^M$$

Prendendo il logaritmo (in base 2) dei termini più a sinistra e più a destra, otteniamo

$$k \log_2(\frac{1}{2}) \leq \log_2(|\mathcal{H}|) + M \log_2(\frac{3}{4})$$

ed isolando M otteniamo (notare che $\log_2(\frac{3}{4})$ è una quantità negativa)

$$M \leq \frac{k+\log_2(|\mathcal{H}|)}{-\log_2(\frac{3}{4})} \leq 2.4(k + \log_2(|\mathcal{H}|))$$