Algorithmisches Lernen
2. Übungsblatt

Adam Duck

8. Mai 2001

2.1 Aufgabe: Zur Klassifizierung von Beispielen steht uns bei den ersten beiden Aufgaben eine Runde von Experten zur Verfügung. Diese Experten können aber gelegentlich irren. Folge davon ist, dass das Verfahren nicht zwangsläufig terminiert. Es also nur Sinn, nach der Zahl der Fehler bis zu einem beliebigen aber festen Zeitpunkt k zu fragen:

Um Fehler abzufangen, variieren wir Weighted Majority wie folgt:

- Zu Beginn weise jedem Experten eine *Glaubwürdigkeit* w_i von 1 zu
- Unsere Hypothese h ist zu jedem Zeitpunkt:

$$h(\vec{x}) = 1 \iff \sum_{E_i(\vec{x})=1} w_i \geq \sum_{E_i(\vec{x})=0} w_i$$

  Dabei bezeichnet $E_i(\vec{x})$ die Klassifizierung, die Experte i bei Beispiel $\vec{x}$ vornimmt.
- Solange ein Gegenbeispiel $\vec{x}$ geliefert wird, aktualisiere die Glaubwürdigkeiten der Experten wie folgt:

$$w_i := \begin{cases} w_i & \text{falls } E_i \text{ bei } \vec{x} \text{ richtig lag} \\ \frac{w_i}{2} & \text{sonst} \end{cases}$$

Zeigen Sie, dass nach k Gegenbeispielen für den Algorithmus auch der beste Experte mindestens $\Omega(k) - \mathcal{O}(\log(n))$ Fehler macht. Obwohl der zuverlässigste Experte nicht bekannt ist, und sich sogar mit wachsendem k ändern kann, wird unser Algorithmus also eine fast genauso gute Klassifikationsleistung zeigen. (8 Punkte)

Lösung:
Sei W die Gesamtglaubwürdigkeit, wobei am Anfang $W = n$ gilt. Wenn wir irren, dann liegt mindestens die Hälfte der Gesamtglaubwürdigkeit der Experten auf der "falschen" Seite – was nicht bedeuten muss, dass das die Hälfte der Experten sind – und das bedeutet, dass sich W bei jedem Gegenbeispiel um mindestens $\frac{1}{4}$ verkleinert. Nach k Gegenbeispielen ist also:

$$W \leq n \cdot \left(\frac{3}{4}\right)^k$$

Dabei hat unser bester Experte m Fehler gemacht ($m \leq k$). Seine Glaubwürdigkeit ist dabei $\geq \left(\frac{1}{2}\right)^m$. Nun ist

$$\begin{aligned}
w &\leq W \\
\left(\frac{1}{2}\right)^m &\leq n \cdot \left(\frac{3}{4}\right)^k \\
m \cdot \log(2) &\geq \log\left(\frac{1}{n} \cdot \left(\frac{4}{3}\right)^k\right) \\
m &\geq -\log(n) + k \cdot \log\left(\frac{4}{3}\right) \\
m &\geq \Omega(k) - \mathcal{O}(\log(n))
\end{aligned}$$

2.2 Aufgabe: In einer zweiten Variation wählen wir in jeder Iteration einen Experten zufällig und übernehmen seine Hypothese. Dabei wählen wir den Experten E_i mit Wahrscheinlichkeit

$$\frac{w_i}{\sum_{j=1}^{n} w_j}$$

Diesmal werden wir auch bei Beispielen, die wir (also unser zufällig gewählte Experte) richtig klassifizieren, die irrenden Experten bestrafen. Klassifiziert Experte E_j das Beispiel $\vec{x}$ falsch, so multiplizieren wir seine Glaubwürdigkeit w_j mit β. Dabei gilt $0 \leq \beta \leq 1$.

Da wir den Experten zufällig wählen, macht es nur Sinn, nach der **erwarteten** Anzahl von Fehlern zu fragen.

Zeige, dass die asymptotische Schranke für die Zahl der Fehler des besten Experten nach k **Gegen**beispielen des Algorithmus aus Aufgabe 1 auch hier gilt. Wie verändern sich die Konstanten für verschiedene Werte von β? (8 Punkte)

Lösung:

Sei F_i der Anteil der falschen Antworten an der Gesamtglaubwürdigkeit beim i-ten Beispiel. Nach k Beispielen ist die erwartete Anzahl von Fehlern

$$M = \sum_{i=1}^{k} F_i$$

Desweiteren ändert sich die Gesamtglaubwürdigkeit beim i-ten Beispiel wie folgt:

$$W_i = W_{i-1} - W_{i-1}F_i(1-\beta) = W_{i-1}(1-(1-\beta)F_i)$$

Damit ergibt sich nach dem k-ten Beispiel

$$W = n \cdot \prod_{i=1}^{k}(1-(1-\beta)F_i)$$

Sei wiederum m die Fehleranzahl des besten Experten, dann gilt:

$$
\begin{aligned}
1 \cdot \beta^m &\leq n \cdot \prod_{i=1}^{k} (1 - F_i(1-\beta)) && \text{1 ist die Glaubwürdigkeit zu Beginn} \\
m \ln(\beta) &\leq \ln(n) + \sum_{i=1}^{k} \ln(1 - F_i(1-\beta)) && \ln(1-x) \leq -x \text{ für } x \leq 1 \\
m \ln(\beta) &\leq \ln(n) + \sum_{i=1}^{k} (-F_i(1-\beta)) \\
m &\geq \frac{\ln(n)}{\ln(\beta)} - \frac{(1-\beta)M}{\ln(\beta)} && \ln(\beta) < 0! \\
m &\geq \frac{(1-\beta)}{\ln\left(\frac{1}{\beta}\right)} M - \frac{1}{\ln\left(\frac{1}{\beta}\right)} \ln(n) \\
m &\geq c_1 M - c_2 \ln(n)
\end{aligned}
$$

Und damit bekommen wir:

$$m \geq \Omega(k) - \mathcal{O}(\log(n))$$

Wenn β kleiner wird, werden sowohl c_1 als auch c_2 größer, was sich dann in mehr Fehlern des besten Experten niederschlägt, denn die Funktion an c_1 wächst linear und die an c_2 logarithmisch. Ein größeres β ist also besser für den Algorithmus.