Boosting
- Formelübersicht -

Alexander Dockhorn

a.dockhorn@qmul.ac.uk

16. März 2021

Diese kurze Übersicht umfässt die in der Lehrprobe zum Thema Boosting verwendeten Formelzeichen, Gleichungen und Algorithmen.

Da diese Übersicht in ihrem Umfang begrenzt ist, möchte ich Sie zudem auf eine vergleichbare Übersicht zur Vorlesung Computational Intelligence in Games (`https://github.com/ADockhorn/CIG-Overview/blob/master/equations.pdf`) verweisen, welche ich im Rahmen meiner Vorlesungsvertretung an der Otto-von-Guericke Universität Magdeburg gemeinsam mit Studierenden erstellte.

Falls Sie Ideen zur Verbesserung dieser Übersicht haben, würde ich mich über eine E-Mail oder einen Pull-Request sehr freuen.

Gleichungen und Herleitungen

Algorithmen

1 Boosting

Symbol	Name	Beschreibung
X	Eingabedaten	Menge der zu klassifizierenden Datenpunkte, $x_i \in X \subset \mathbb{R}^n$
y	Ausgabedaten	Vektor der erwartenden Ausgaben, $y_i \in -1, 1$
(x_i, y_i)	Trainingsbeispiel	ein Trainingsbeispiel
D	Trainingsdaten	Menge an Trainingbeispielen $\{(x_i, y_i)\}^n$
t	Zeitpunkt	Zeitpunkt des Trainingsprozesses bzw. Anzahl von Klassifikatoren im Ensemble
h_t	Klassifikator	einzelner Klassifikator, welcher zur Ensemblebildung verwendet wird
$h_t(x)$	Ausgabe des Klassifikators	dem Datenpunkt x zugeordnetes Label, $h_t(x) \in -1, 1$
α_t	Gewicht eines Klassifikators	Gewichtung zur Aggregation von Klassifikatoren des Ensembles
$H_t(x)$	Ensemble	Ensemble welches mehrere Klassifikatoren aggregiert
$\mathcal{L}(H_t \vert y)$	Fehlerfunktion	Funktion zum Messen des Ensemblefehlers
$I(x \neq y)$	Indikatorfunktion	Verwendet zur Bewertung des Ensemblefehlers bezüglich der erwünschten Klassifikation y (siehe Gleichung 4)
ε	gewichteter Fehler	findet Verwendung in der Herleitung von Adaboost

1.1 Adaptive Boosting (Adaboost)

Exponentieller Fehler über die gewichteten Trainingsdaten

$$\mathcal{L}(H_t \mid y) = \sum_{i}^{|D|} w_i^t e^{-\frac{1}{2} y_i H_t(x_i)} \tag{1}$$

Indikatorfunktion zur Erkennung falsch klassifizierter Trainingsbeispiele

$$I(x \neq y) = \begin{cases} 1 & , x \neq y \\ 0 & , \text{sonst} \end{cases} \tag{2}$$

Ausgabe des Ensembles

$$H_t(x) = \sum_{i=1}^{t} \alpha_i h_i(x) \tag{3}$$

Label Zuweisung des Ensembles

$$H_t(x) = \operatorname{sign}\Big(\sum_{i=1}^{t} \alpha_i h_i(x)\Big) \tag{4}$$

Herleitung zur Wahl des optimalen Klassifikators

$$\begin{aligned}
&\mathcal{L}(H_{t+1}|y) \\
&= \sum_{i}^{|D|} w_i^t e^{-\frac{1}{2}y_i\alpha_{t+1}h_{t+1}(x_i)} && \Big|\ \sum_{alle} = \sum_{falsch} + \sum_{richtig} \\
&= \underbrace{\sum_{i:h_{t+1}(x_i)\neq y_i}^{|D|} w_i^t e^{-\frac{1}{2}y_i\alpha_{t+1}h_{t+1}(x_i)}}_{\text{falsch klassifizierte Punkte}} + \underbrace{\sum_{i:h_{t+1}(x_i)= y_i}^{|D|} w_i^t e^{-\frac{1}{2}y_i\alpha_{t+1}h_{t+1}(x_i)}}_{\text{richtig klassifizierte Punkte}} && \Big|\ y_i, h(x_i) \in \{-1,+1\} \\
&= \sum_{i:h_{t+1}(x_i)\neq y_i}^{|D|} w_i^t e^{\frac{\alpha_{t+1}}{2}} + \sum_{i:h_{t+1}(x_i)= y_i}^{|D|} w_i^t e^{-\frac{\alpha_{t+1}}{2}} && \Big|\ I(x\neq y) = \begin{cases} 1 & ,x\neq y \\ 0 & ,\text{sonst} \end{cases} \\
&= e^{\frac{\alpha_{t+1}}{2}} \sum_{i}^{|D|} w_i^t I(h_{t+1}(x_i)\neq y_i) \\
&\quad + \underbrace{e^{-\frac{\alpha_{t+1}}{2}} \sum_{i}^{|D|} w_i^t}_{\text{alle Punkte}} - \underbrace{e^{-\frac{\alpha_{t+1}}{2}} \sum_{i}^{|D|} w_i^t I(h_{t+1}(x_i)\neq y_i)}_{\text{falsch klassifizierte Punkte}} && \Big|\ \sum_{richtig} = \sum_{alle} - \sum_{falsch} \\
&= (e^{\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}}) \sum_{i}^{|D|} w_i^t I(h_{t+1}(x_i), y_i) + e^{-\frac{\alpha_{t+1}}{2}} \sum_{i}^{|D|} w_i^t
\end{aligned}$$

Wir wählen zu jedem Zeitpunkt den Klassifikator, welcher den exponentiellen Fehler über alle Trainingsbeispiele minimiert. Die Herleitung zeigt, dass die optimale Wahl des Klassifikators lediglich abhängig von der Minimierung des gewichteten Fehlers ist, da alle anderen Komponenten der Funktion konstant sind.
Resultat: wähle den Klassifikator zur Minimierung des gewichteten Fehlers

Herleitung zur Wahl des optimalen Klassifikatorgewichts

Nach der Wahl des optimalen Klassifikators h_{t+1} bestimmen wir dessen Gewicht, indem wir die Fehlerfunktion nach α ableiten.

$$\mathcal{L}(H_{t+1}|y) = (e^{\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}}) \sum_{i}^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) - e^{-\frac{\alpha_{t+1}}{2}} \underbrace{\sum_{i}^{|D|} w_i^t}_{=1}$$

$$\frac{d\mathcal{L}(H_{t+1}|y)}{d\alpha} = \frac{1}{2}(e^{\frac{\alpha_{t+1}}{2}} + e^{-\frac{\alpha_{t+1}}{2}}) \sum_{i}^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) - \frac{1}{2} e^{-\frac{\alpha_{t+1}}{2}}$$

Zur Vereinfachung substituieren wir $\varepsilon = \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)$

$$\frac{d\mathcal{L}(H_{t+1}|y)}{d\alpha} = \frac{1}{2}(e^{\frac{\alpha_{t+1}}{2}} + e^{-\frac{\alpha_{t+1}}{2}})\varepsilon - \frac{1}{2} e^{-\frac{\alpha_{t+1}}{2}}$$

Suche den Extrempunkt bezüglich α, indem die Ableitung gleich 0 gesetzt wird.

$$\begin{aligned}
\frac{1}{2}(e^{\frac{\alpha_{t+1}}{2}} + e^{-\frac{\alpha_{t+1}}{2}})\varepsilon - \frac{1}{2} e^{-\frac{\alpha_{t+1}}{2}} &= 0 && \Big| : \frac{1}{2} \\
e^{\frac{\alpha_{t+1}}{2}}\varepsilon + e^{-\frac{\alpha_{t+1}}{2}}\varepsilon - e^{-\frac{\alpha_{t+1}}{2}} &= 0 && \Big| - (e^{-\frac{\alpha_{t+1}}{2}}\varepsilon - e^{-\frac{\alpha_{t+1}}{2}}) \\
e^{\frac{\alpha_{t+1}}{2}}\varepsilon &= e^{-\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}}\varepsilon \\
e^{\frac{\alpha_{t+1}}{2}}\varepsilon &= e^{-\frac{\alpha_{t+1}}{2}}(1-\varepsilon) && \Big| \ln; \qquad \ln(xy) = \ln(x) + \ln(y) \\
\frac{\alpha_{t+1}}{2} + \ln(\varepsilon) &= -\frac{\alpha_{t+1}}{2} + \ln(1-\varepsilon) && \Big| + \frac{\alpha_{t+1}}{2} - \ln\varepsilon \\
\alpha_{t+1} &= \ln(1-\varepsilon) - \ln(\varepsilon) && \Big| \ln(x) - \ln(y) = \ln(\frac{x}{y}) \\
\alpha_{t+1} &= \ln\left(\frac{1-\varepsilon}{\varepsilon}\right)
\end{aligned}$$

Resultat: die Schrittweite ist abhängig von dem gewichteten Fehler des Klassifikators

Herleitung zur Aktualisierung der Trainingsdatengewichtung

$$\begin{aligned}
\mathcal{L}(H_{t+1}|y) &= \sum_{i}^{|D|} e^{-\frac{1}{2}y_i H_{t+1}(x_i)} && \Big|\ H_{t+1}(x_i) = H_t(x_i) + \alpha_{t+1}h_{t+1}(x_i) \\
&= \sum_{i}^{|D|} e^{-\frac{1}{2}y_i H_t(x_i) - \frac{1}{2}y_i \alpha_{t+1} h_{t+1}(x_i)} && \Big|\ e^{a+b} = e^a e^b \\
&= \sum_{i}^{|D|} e^{-\frac{1}{2}y_i H_t(x_i)} e^{-\frac{1}{2}y_i \alpha_{t+1} h_{t+1}(x_i)} && \Big|\ w_i^t = e^{-\frac{1}{2}y_i H_t(x_i)} \\
&= \sum_{i}^{|D|} w_i^t e^{-\frac{1}{2}y_i \alpha_{t+1} h_{t+1}(x_i)} && \Big|\ \Rightarrow w_i^0 = \frac{1}{|D|}; \quad w_i^{t+1} = w_i^t \cdot e^{-\frac{1}{2}\alpha_t y_i h_t(x_i)}
\end{aligned}$$

Resultat: das Gewicht eines Datenpunktes ist proportional zum Fehler des aktuellen Ensembles. Ausgehend von einer Gleichverteilung der Trainingsdatengewichtung zu Beginn des Trainingsprozesses, kann eine rekursive Aktualisierung der Gewichte bestimmt werden.

Algorithm 1: Adaboost: *(Hinweis: Der von Freund und Schapire vorgestellte Adaboost Algorithmus verwendet Entscheidungssstümpfe als Klassifikatoren. Es können auch andere Klassifikatoren verwendet werden, insofern ihre Ausgabe auf die Menge* $\{-1, 1\}$ *beschränkt bleibt und sie bezüglich des gewichteten Fehlers optimiert werden können.*

Input: Trainingsdaten D, Anzahl an Zeitschritten T

Setze $w_i^0 = \frac{1}{|D|}$ für alle Trainingsbeispiele in D

Initialisiere H_0 als leeres Ensemble

for $t \leftarrow 0$ **to** T **do**

- $h_{t+1} \leftarrow$ Lerne Klassifikator zur Minimierung des gewichteten Fehlers
- Bestimme den gewichteten Fehler $\varepsilon = \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)$
- $a_{t+1} \leftarrow \ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$
- Füge $a_{t+1}h_{t+1}$ dem Ensemble H_t hinzu.
- $w_i^{t+1} \leftarrow w_i^t \cdot e^{-\frac{1}{2}\alpha_t y_i h_t(x_i)}$

Output: Ensemble Klassifikator H_T