AGML, Sommersemester 2020 Vorlesung 16 (5 Seiten)

Entscheidungsbäume

Anne Driemel Letzte Aktualisierung: 22. Juni 2020

Entscheidungsbäume sind eine beliebte Form um eine Klassifizierung anhand eine Reihe von Tests darzustellen. Damit kann zum Beispiel auf kompakte Weise dargestellt werden, ob eine Person, die eine bestimmte Kombination von Krankheitssymptomen vorweist, einen Arzt aufsuchen sollte oder sich in häusliche Quarantäne begeben sollte.[1] Entscheidungsbäume werden in der Praxis oft per Hand von einem Experten erstellt, zum Beispiel im Rahmen einer Risikoanalyse.

Im Maschinellen Lernen werden Entscheidungsbäume genutzt um komplexe Kompositionen von einfachen Hypothesen darzustellen. Oft wird als Basis die Klasse der Schwellenwertfunktionen genutzt, also Halbräume die durch achsenparallele Hyperebenen beschränkt sind. Denkbar sind aber auch beliebige Halbräume als Basis.

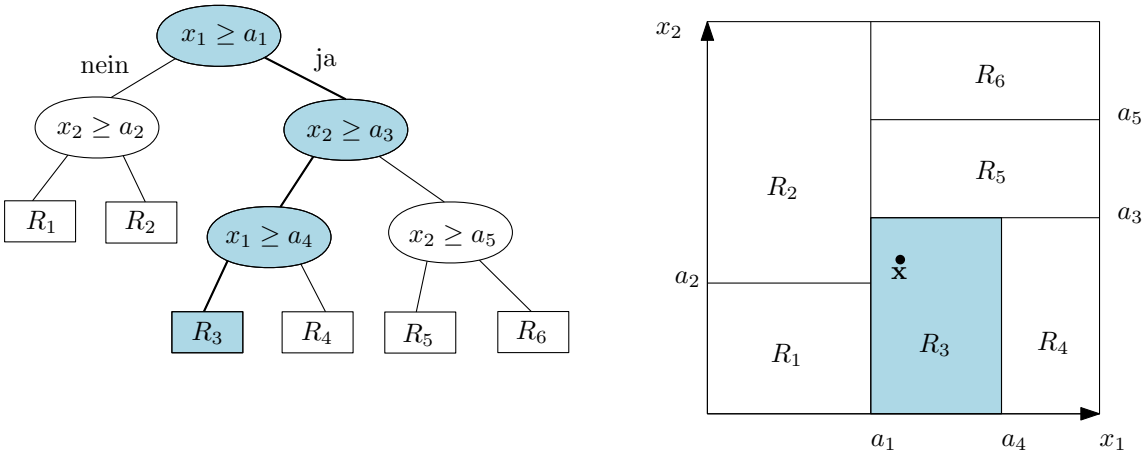

Abbildung 1: Beispiel für die rekursive Partitionierung der Grundmenge $[0,1]^2$ durch einen Entscheidungsbaum mit der Klasse der Schwellenwertfunktion als Basisklasse.

Diese Funktionen werden nun in Form eines binären Baumes rekursiv miteinander kombiniert. Als Ergebnis entsteht eine rekursive Partitionierung der Grundmenge. Jedem Blattknoten ist das Label einer Klasse zugewiesen. Um einen Punkt der Grundmenge zu klassifizieren, folgt man einem Pfad von der Wurzel bis zu dem Blatt, das den Punkt $\mathbf{x}$ enthält und gibt dann das entsprechende Label aus. Siehe Abbildung 1 für ein Beispiel einer Partitionierung für den Fall, dass die Grundmenge $[0,1]^2$ ist.

Der Vorteil einer achsenparallelen Partitionierung gegenüber einer Partitionierung mit Halbräumen, ist, dass jeder Knoten des Entscheidungsbaumes einem Test bezüglich einer festen Komponente des Feature-Vektors darstellt. Zum Beispiel könnte eine bestimmte Komponente darstellen ob und wie stark ein bestimmtes Krankheitssymptom bei einer Person aufgetreten ist. Somit ist die vom Lernalgorithmus berechnete Hypothese von Menschen besser interpretierbar. Daher werden Schwellwertfunktionen auch im Maschinellen Lernen in der Praxis manchmal gegenüber allgemeinen Halbräumen bevorzugt.

[1] Siehe hier für das Corona-Virus
https://www.zeit.de/wissen/gesundheit/2020-03/Coronavirus-Entscheidungshilfe-2020-03-31.pdf

1 Hypothesenklasse

Formal kann man die resultierende Hypothese wie folgt definieren. Sei T ein binärer Baum mit k inneren Knoten und Wurzel w, wobei jedem inneren Knoten v eine Hypothese $h_v \in \mathcal{H}$ und jedem Blattknoten v ein Label $\ell_v \in \{+1, -1\}$ zugewiesen ist. Sei v ein innerer Knoten und sei v_L das linke Kind von v in T und sei v_R das rechte Kind von v in T. Definiere die Funktion $g_v : X \to \{+1, -1\}$ für einen inneren Knoten v mit

$$g_v(\mathbf{x}) = \begin{cases} g_{v_R}(\mathbf{x}) & \text{falls } h_v(\mathbf{x}) = 1 \\ g_{v_L}(\mathbf{x}) & \text{sonst} \end{cases} \tag{1}$$

Falls v ein Blattknoten ist, dann definieren wir $g_v(\mathbf{x}) = \ell_v$.

Die Hypothese g_w, definiert durch den Baum T und den assoziierten Hypothesen an den inneren Knoten sowie den Labelzuweisungen an den Blättern von T, stellt dann einen Entscheidungsbaum mit Basisklasse $\mathcal{H}$ dar. Dies ist eine Komposition, ähnlich wie wir sie in der letzten Vorlesung definiert haben, wobei der Baum T die Kompositionsfunktion definiert. Der einzige Unterschied ist, dass wir zusätzlich zu der Basisklasse auch die Labelzuweisungen an den Blättern haben.

Definition 16.1. *Sei B_k die Menge der gewurzelten binären Bäume mit k inneren Knoten und $k+1$ Blättern, wobei jeder innere Knoten v ein linkes Kind v_L und ein rechtes Kind v_R hat, und genau einen Elternknoten.*

Definition 16.2 (Entscheidungsbaum)**.** *Sei $\mathcal{H}$ eine Hypothesenklassen mit Grundmenge X und sei $\mathcal{R}$ das zugehörige Mengensystem. Sei $\mathcal{H}_{B_k}$ die Hypothesenklasse aller Funktionen $g_w : X \to \{+1, -1\}$ definiert wie in (1) durch*

(i) einen binären Baum $T \in B_k$ mit inneren Knoten $v_1, \ldots, v_k$ und Blättern $b_1, \ldots b_{k+1}$,

(ii) Hypothesen $h_1, \ldots, h_k \in \mathcal{H}$ und Labels $\ell_1, \ldots \ell_{k+1} \in \{+1, -1\}$

Wir legen dabei fest, dass $w = v_1$ die Wurzel des Baumes T ist und dass, für $1 \leq i \leq k$, die Hypothese h_i dem inneren Knoten v_i zugewiesen ist, sowie dass, für $1 \leq i \leq k+1$, das Label ℓ_i dem Blatt b_i zugewiesen ist.

Das folgende Lemma wird uns helfen, eine obere Schranke für die VC-Dimension der Hypothesenklasse der Entscheidungsbäume mit k inneren Knoten zu zeigen.

Lemma 16.3. *Für jede natürliche Zahl $k \geq 1$ ist $\left|B_k\right| \leq k!$*

Beweis. Wir zeigen dies durch Induktion. Sei $k = 1$. In diesem Fall gibt es nur einen Baum in B_k, nämlich die Wurzel selbst mit zwei Blättern. Also ist $\left|B_1\right| = 1 = 1!$ korrekt.

Sei $k > 1$. Wir können jeden Baum $T \in B_k$ aus einem Baum $T' \in B_{k-1}$ erzeugen, indem wir in T' einen Blattknoten entfernen und an derselben Stelle einen Knoten mit zwei neuen Blattknoten als Kindern hinzufügen. Tatsächlich hat der so erzeugte Baum k innere Knoten und $k+1$ Blattknoten (ein Blattknoten wurde entfernt und zwei neue Blattknoten sind hinzugekommen). Die Eigenschaft, dass jeder innere Knoten zwei Kinder hat bleibt dadurch gleichermaßen unberührt.

Für einen festen Baum $T' \in B_{k-1}$ gibt es genau k verschiedenen Möglichkeiten solch einen Baum in B_k zu erzeugen, da T' genau k Blattknoten hat. Also ist

$$\left|B_k\right| \leq k \cdot \left|B_{k-1}\right|$$

Nun können wir die Induktionsannahme für $\left|B_{k-1}\right|$ einsetzen und erhalten

$$\left|B_k\right| \leq k \cdot \left|B_{k-1}\right| \leq k \cdot (k-1)! \leq k!$$

(Es kann passieren, dass der gleiche Baum in B_k durch zwei verschiedene Bäume in B_{k-1} erzeugt wird, aber das stört uns nicht, da wir nur eine obere Schranke zeigen wollen.) □

2 VC-Dimension

Satz 16.4. *Sei $\mathcal{H}$ eine Hypothesenklasse mit Grundmenge X und VC-Dimension* d *mit* $\mathrm{d} < \infty$. *Sei $k \geq 2$ eine natürliche Zahl. Die VC-Dimension von $\mathcal{H}_{B_k}$ ist höchstens* $20\mathrm{d}k\ln(10k)$.

Beweis. Wir nutzen einen ähnlichen Beweis wie in der letzten Vorlesung. Diesmal müssen wir die Anzahl der verschiedenen Kompositionsfunktionen miteinbeziehen, die durch verschiedene Bäume in B_k entstehen können.

Sei $A \subseteq X$ eine Menge, die von $\mathcal{H}_{B_k}$ aufgespalten wird und sei $t = \big|A\big|$. Wir wollen wieder eine obere Schranke für die Anzahl Hypothesen in $\mathcal{H}_{B_k}|_A$ finden, und nutzen, dass $2^t \leq \big|\mathcal{H}_{B_k}|_A\big|$.

Laut Lemma 16.3 gibt es höchstens $k!$ verschiedene Bäume in B_k. Weiter müssen wir k Hypothesen aus $\mathcal{H}|_A$ auswählen. Für die Zuweisung der Labels an die $k+1$ Blätter des Baumes gibt es 2^{k+1} Möglichkeiten. Damit wäre eine Hypothese in $\mathcal{H}_{B_k}|_A$ eindeutig identifiziert.

Wir können also die Anzahl der Hypothesen in $\mathcal{H}_{B_k}|_A$ abschätzen indem wir die Anzahl der verschiedenen Bäume, die Anzahl der verschiedenen Label-Zuweisungen an die Blätter und die Anzahl der Möglichkeiten, k Hypothesen an die inneren Knoten zuzuweisen, miteinander multiplizieren. Es ergibt sich also

$$\big|\mathcal{H}_{B_k}|_A\big| \leq k! \cdot 2^{k+1} \cdot \big|\mathcal{H}|_A\big|^k \leq k! \cdot 2^{k+1} \cdot (\Pi_{\mathcal{H}}(t))^k \leq k! \cdot 2^{k+1} \cdot \left(\frac{\mathrm{e}t}{\mathrm{d}}\right)^{\mathrm{d}k}$$

wobei die letzte Ungleichung wieder aus dem Wachstumslemma folgt. Da alle 2^t verschiedenen Teilmengen von A dargestellt werden, ergibt sich ähnlich wie zuvor, dass

$$2^t \leq k! \cdot 2^{k+1} \cdot \left(\frac{\mathrm{e}t}{\mathrm{d}}\right)^{\mathrm{d}k} \leq k^k \cdot k^{k+1} \cdot \left(\frac{t}{\mathrm{d}}\right)^{2\mathrm{d}k} \leq \left(\frac{t}{\mathrm{d}}\right)^{k+k+1+2\mathrm{d}k} \leq \left(\frac{t}{\mathrm{d}}\right)^{5\mathrm{d}k}$$

wobei wir hier annehmen, dass $t \geq \mathrm{de}$, und dass $t \geq \mathrm{d}k$, sonst ist die Aussage trivial erfüllt. Wir können nun beide Seiten logarithmieren und erhalten

$$t\ln 2 \quad \leq 5\mathrm{d}k\ln\tfrac{t}{\mathrm{d}}$$

Da $0.5 \leq \ln 2$, ist also

$$\frac{t}{\mathrm{d}} \quad \leq 10k\ln\tfrac{t}{\mathrm{d}}$$

In Lemma 15.6 hatten wir gezeigt, dass für jedes $x > 0$ und $u \in \mathbb{R}$ gilt

$$x \leq u\ln x \implies x \leq 2u\ln u$$

Das können wir nun mit $x = \frac{t}{\mathrm{d}}$ und $u = 10\mathrm{d}k$ anwenden und erhalten

$$t \leq 20\mathrm{d}k\ln(10k)$$

Da dies für jede Menge A gilt, die aufgespalten wird, folgt direkt die obere Schranke für die VC-Dimension. □

Satz 16.4 besagt, dass die VC-Dimension von Entscheidungsbäumen nur von der Anzahl der inneren Knoten k und von der VC-Dimension der Basisklasse abhängt. Gleichzeitig kann man für jede Menge $S \subseteq \mathbb{R}$ der Größe m einen Entscheidungsbaum mit $k = m - 1$ inneren Knoten finden, der Trainingsfehler null hat. Das heißt, für $k \to \infty$ ist die VC-Dimension von $\mathcal{H}_{B_k}$ im schlimmsten Fall unbeschränkt.

3 Lernalgorithmen

Da bei Entscheidungsbäumen, ähnlich wie beim Boosting, die VC-Dimension mit der Komplexität der Hypothesenklasse steigt, besteht auch hier die Gefahr des Overfittings. Aus diesem Grund will man in der Praxis die Anzahl der inneren Knoten des Baumes beschränken.

Das ist aber oft nicht effizient möglich. Es ist zum Beispiel NP-schwer für eine gegebene Menge $S \subseteq \mathbb{R}^3 \times \{+1, -1\}$ und einen Parameter $k \in \mathbb{N}$ einen optimalen Entscheidungsbaum mit k inneren Knoten zu finden, wenn als Basisklasse die Klasse der Halbräume angenommen wird. Das gilt selbst in dem vergleichsweise einfachen Fall, dass die Labels aus der Menge $\{+1, -1\}$ kommen und die Dimension der Grundmenge $d = 3$ ist.

Daher werden in der Praxis Entscheidungsbäume meist heuristisch optimiert, indem die Knoten nacheinander hinzugefügt werden, wobei in jedem Schritt die Zielfunktion lokal optimiert wird. Der Algorithmus muss lokal entscheiden, welcher Knoten hinzugefügt wird und nimmt meist den Knoten, dessen assoziierte Trainingsmenge den größten Trainingsfehler hat. Denkbar ist auch, erst einen größeren Baum zu bauen und dann heuristisch Unterbäume zu entfernen. Das Problem dabei ist aber, dass selbst die erste Hypothese im Wurzelknoten die optimale Lösung blockieren kann.

3.1 Greedy-Algorithmus

Wir wollen trotzdem eine einfache Variante dieses Greedy-Algorithmus genauer definieren. Der Algorithmus bekommt als Eingabe einen Parameter k und eine Datenpunkt/Label-Menge $S = ((x_1, y_1), \ldots, (x_m, y_m))$. Jeder Blattknoten v hat eine assoziierte Menge $S_v \subseteq S$, welche nur für die Konstruktion des Baumes verwendet wird.

`decisionTree`**(k,S)**
1. Initialisiere T mit einem Blattknoten v
2. `initLeaf`(v,S)
3. **for** i **in** $1 \ldots k$ **do**
4. Finde Blattknoten v in T mit größtem Klassifizierungsfehler $err_{S_v}(T)$
5. `split`(v, T)
6. **for** v **in** Menge der Blattknoten von T **do**
7. Entferne die Menge S_v von dem Blattknoten v
8. Gebe den Baum T zurück

`split`**(v,T)**
1. Berechne eine Hypothese $h \in \mathcal{H}$, welche $err_{S_v}(h)$ minimiert
2. Assoziiere mit v die Hypothese h
3. Entferne die Menge S_v von v
4. Füge v_L als linkes Kind von v zu T hinzu
5. Sei $S_{v_L} = \{ (x, y) \in S_v \mid h(x) = -1 \}$
6. `initLeaf`(v_L, S_{v_L})
7. Füge v_R als rechtes Kind von v zu T hinzu
8. Sei $S_{v_R} = \{ (x, y) \in S_v \mid h(x) = +1 \}$
9. `initLeaf`(v_L, S_{v_L})

`initLeaf(v,S)`
1. Assoziiere mit v die Menge S
2. Assoziiere mit v das Label welches unter den Punkten in S_v am meisten vertreten ist

3.2 Random-Forest-Algorithmus

Um die Stabilität des Lernalgorithmus zu verbessen, werden Entscheidungsbäume oft auf zufällig gewählten Untermengen der Trainingsmenge heuristisch berechnet und die entstandenen Hypothesen mit zufällig gewählten Gewichten kombiniert. Dieser Lernalgorithmus wird als *Random-Forest-Algorithmus* bezeichnet. Die VC-Dimension kann hier wieder durch die Linearkombination der einzelnen Hypothesen wachsen. Allerdings tritt bei Random Forests das Phänomen des Overfittings in der Praxis fast nie auf.

Referenzen

- Understanding Machine Learning, Kapitel 18 (Decision Trees)
- Foundations of Machine Learning, Kapitel 9.3.3 (Decision Trees)
- Michael T. Goodrich , Vincent Mirelli , Mark Orletsky , Jeffery Salowe, "Decision Tree Construction in Fixed Dimensions: Being Global is Hard but Local Greed is Good" Technical Report TR-95-1, Johns Hopkins University, 1995.