Einführung in die Künstliche Intelligenz

Übungszettel 3

Prof. Dr. Claudia Schon

C.Schon@hochschule-trier.de

Fachbereich Informatik
Hochschule Trier

1 min, arg min, max, und arg max

Im Folgenden betrachten wir jeweils eine Funktion $f : X \to \mathbb{R}$ mit entsprechendem Definitionsbereich X. Wir möchten für diese Funktionen jeweils $\min f(x)$ (oder $\max f(x)$) sowie $\arg\min f(x)$ (oder $\arg\max f(x)$) bestimmen.

Beispiel:
Wir bestimmen $\min f(x)$ und die Menge der Argumente, für die das Minimum erreicht wird, (also $\arg\min f(x)$), für die Funktion $f(x) = |x - 1|, \quad X = \{-2, -1, 0, 1, 2, 3\}$:

Wir berechnen $f(x)$ für alle $x \in X$:

$$\begin{aligned} f(-2) &= |-2-1| = 3 \\ f(-1) &= |-1-1| = 2 \\ f(0) &= |0-1| = 1 \\ f(1) &= |1-1| = 0 \\ f(2) &= |2-1| = 1 \\ f(3) &= |3-1| = 2 \end{aligned}$$

Das Minimum ist also

$$\min f(x) = 0,$$

und es wird erreicht bei

$$\arg\min f(x) = \{1\}.$$

a) Bestimmen Sie jeweils $\min f(x)$ und die Menge der Argumente, für die das Minimum erreicht wird, also $\arg\min f(x)$, für die folgenden Funktionen:

   a) $f(x) = x^2, \quad X = \{-2, -1, 0, 1, 2\}$

   b) $f(x) = |x - 3|, \quad X = \{1, 2, 3, 4, 5\}$

   c) $f(x) = -x^3, \quad X = \{-2, -1, 0, 1, 2\}$

b) Bestimmen Sie jeweils $\max f(x)$ und die Menge der Argumente, für die das Minimum erreicht wird, also $\arg\max f(x)$, für die folgenden Funktionen:

   a) $f(x) = 2x - 5, \quad X = \{0, 1, 2, 3\}$

   b) $f(x) = -x^2 + 4x, \quad X = \{0, 1, 2, 3, 4\}$

   c) $f(x) = |x|, \quad X = \{-3, -2, -1, 0, 1, 2, 3\}$

2 Entscheidungsbaum erstellen[1]

Ein Tennistrainer möchte eine Regel aufstellen, um zu entscheiden, ob Tennis gespielt werden sollte oder nicht. Dazu hat er über 14 Tage Wetterdaten gesammelt und für jeden Tag festgehalten, ob gespielt wurde. Die Daten sind in der folgenden Tabelle dargestellt:

Outlook	**Humidity**	**Wind**	**PlayTennis**
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No

(a) Bestimmen Sie das beste Attribut für die Wurzel des Entscheidungsbaums, indem Sie den *Information Gain* für jedes Attribut berechnen.

(b) Konstruieren Sie schrittweise einen Entscheidungsbaum für die obigen Trainingsdaten, indem Sie den in der Vorlesung vorgestellten Algorithmus anwenden.

(c) Wir möchten einen Datensatz zur Beispielmenge hinzufügen, so dass im Entscheidungsbaum zur erweiterten Trainingsmenge ein anderes Attribut an der Wurzel steht. Welchen Datensatz fügen Sie hinzu? Beschreiben Sie Ihr Vorgehen bei der Auswahl des Datensatzes.

[1] Aufgabe basiert auf einem Beispiel aus: *Mitchell, T. M. (1997). Machine learning (Vol. 1). McGraw-hill New York.*

3 Entscheidungsbaum in KNIME erstellen

KNIME ist eine grafische Open-Source-Plattform für Datenanalyse, Machine Learning und Data Mining. In dieser Übung nutzen Sie KNIME, um einen Entscheidungsbaum für die Trainingsdaten aus der vorherigen Aufgabe zu erstellen.

KNIME ist auf den Poolrechnern der Hochschule bereits installiert. Alternativ kann es kostenlos unter `https://www.knime.com` heruntergeladen werden.

(a) Erstellen Sie nun mit KNIME einen Entscheidungsbaum, der vorhersagt, ob Tennis gespielt wird, basierend auf Wetterdaten aus der vorherigen Aufgabe. Gehen Sie dazu wie folgt vor:

1. Laden Sie die Datei `play_tennis_data_small.xlsx` aus Stud.IP herunter.
2. Öffnen Sie KNIME und erstellen Sie über das `"+"`-Symbol einen neuen Workflow (z. B. mit dem Namen `PlayTennis_Entscheidungsbaum`).
3. Ziehen Sie die heruntergeladene `.xlsx`-Datei in den Arbeitsbereich von KNIME. Dadurch wird automatisch ein *Excel Reader* Knoten erstellt.
4. Doppelklicken Sie auf den *Excel Reader* Knoten, um die Vorschau der Daten anzuzeigen. Klicken Sie dann einmal auf den Knoten und führen Sie ihn über *Execute* aus, um die Daten zu laden.
5. Suchen Sie im linken Leiste von KNIME unter *Nodes* nach dem *Decision Tree Learner* Knoten, fügen Sie ihn hinzu und verbinden Sie ihn mit dem Ausgang des *Excel Reader* Knotens.
6. Doppelklicken Sie auf den *Decision Tree Learner* Knoten und nehmen Sie unter dem Reiter *Options* folgende Einstellungen vor:
   - Class column: *PlayTennis*
   - Quality measure: *Gain ratio*
   - Pruning method: *No pruning*
   - Reduced Error Pruning: *ausgeschaltet*
   - Min number records per node: *1*
   - Number of records to store for view: *10000*
   - Average split point: *ausgeschaltet*
   - Number threads: *8*
   - Skip nominal columns without domain information: *ausgeschaltet*
7. Führen Sie den *Decision Tree Learner* Knoten über *Execute* aus.

8. Fügen Sie nun den *Decision Tree to Image* Knoten hinzu und verbinden Sie den oberen Eingang mit dem Ausgang des *Decision Tree Learner* Knotens.
9. Führen Sie auch diesen Knoten über *Execute* aus und öffnen Sie die Bildanzeige, um den erstellten Entscheidungsbaum zu betrachten.

Vergleichen Sie den so erstellten Entscheidungsbaum mit dem, den Sie in der vorherigen Aufgabe berechnet haben.

Hinweis: Wenn beim Ausführen eines Knotens ein gelbes Warndreieck erscheint, überprüfen Sie die Einstellungen und führen Sie den vorherigen Knoten ggf. erneut aus.

(b) Erstellen Sie auch für Ihre um ein Beispiel erweiterte Trainingsbeispielmenge mit KNIME einen Entscheidungsbaum (siehe Aufgabenteil (c) der vorherigen Aufgabe).

4 Gini-Index als Kriterium für Entscheidungsbäume

Der Gini-Index ist ein alternatives Kriterium zur Auswahl des besten Attributs in einem Entscheidungsbaum. Er misst, wie „rein“ eine Datenmenge im Hinblick auf die Klassenzugehörigkeit ist.

Definition des Gini-Index

Für eine Menge von Trainingsbeispielen S mit k Klassen ist der Gini-Wert definiert als:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{k} p_i^2$$

Dabei ist p_i der Anteil der Beispiele in S, die zur Klasse i gehören.

Dann wird der Gini-Index für das Attribut A wie folgt berechnet:

$$\text{Gini-Index}(S, A) = \sum_{v \in \mathit{Values}(A)} \frac{|S_v|}{|S|} \cdot \text{Gini}(S_v)$$

Wobei $S_v = \{s \in S \mid A(s) = v\}$ (S_v ist also die Menge der Beispiele, die für das Attribut A den Wert v haben).

Das Attribut mit dem *kleinsten* Gini-Index wird bevorzugt, da es zu einer möglichst reinen Aufteilung der Daten führt.

In den folgenden Aufgabenteilen betrachten wir nur binäre Klassifikation.

(a) Gegeben sei ein Datensatz S mit nur zwei Klassen: *positiv* und *negativ*. Wie in der Vorlesung bezeichnen wir mit p die Anzahl der positiven Beispiele und mit n die Anzahl der negativen Beispiele.

Wir definieren den Anteil positiver Beispiele als

$$q = \frac{p}{p+n}$$

Drücken Sie den Gini-Wert $\text{Gini}(S)$ in Abhängigkeit von q aus und vereinfachen Sie den Ausdruck.

Interpretieren Sie den Wert in Bezug auf die Reinheit der Daten. Was passiert, wenn $q = 0$, $q = 1$ oder $q = 0{,}5$?

(b) Berechnen Sie den Gini-Wert $\text{Gini}(S)$ für die gesamte Trainingsmenge aus Aufgabe 2 in Bezug auf die Zielvariable *PlayTennis*.

(c) Berechnen Sie den Gini-Index Gini-Index(S, A) für jedes der drei Attribute: *Outlook*, *Humidity* und *Wind*. Welches Attribut besitzt den geringsten Gini-Index?

(d) Überlegen Sie, welchen Wert Gini-Index(S, A) annimmt, wenn das Attribut A sich wie eine *Kundennummer* oder ein *Datum* verhält, d. h. für jedes Beispiel einen eindeutigen Wert hat.

(e) Der Gini-Wert Gini(S) kann auch als die Wahrscheinlichkeit interpretiert werden, dass eine zufällig ausgewählte Instanz aus der Menge S *falsch klassifiziert* wird, wenn man sie gemäß der Klassenverteilung in S zufällig einer Klasse zuweist.

Begründen Sie, warum diese Interpretation korrekt ist.

Hinweis: Gehen Sie davon aus, dass Sie eine Instanz zufällig aus S auswählen und sie dann *ebenfalls zufällig*, basierend auf den Anteilen der Klassen in S, klassifizieren.