Einführung in die Künstliche Intelligenz

Übungszettel 4

Prof. Dr. Claudia Schon

C.Schon@hochschule-trier.de

Fachbereich Informatik
Hochschule Trier

1 Gini-Index als Kriterium für Entscheidungsbäume

Hinweis:

- Diese Aufgabe ist erneut auf dem Übungszettel, da die Aufgabe in der letzten Übungssstunde nicht in allen Gruppen besprochen werden konnte.
- Bei Übungsgruppen, in denen die Aufgabe besprochen werden konnte, können Sie gerne in der nächsten Stunde noch Fragen zu dieser Aufgabe oder zum letzten Übungszettel stellen.

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 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 Gini(S) für die gesamte Trainingsmenge aus Aufgabe 2 (Übungszettel 3) 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.

2 Klassifikatoren vergleichen

Wir möchten erkennen, ob es es sich bei einer gegebenen E-Mail um Spam (positive Klasse, kurz *pos*) oder nicht um Spam (negative Klasse, kurz *neg*) handelt. Wir haben zwei Klassifikatoren A und B trainiert und sie auf einen Testdatensatz angewendet. Die folgende Tabelle zeigt die Ergebnisse beider Klassifikatoren auf diesem Datensatz:

Text	Gold-Label	Klassifikation durch A	Klassifikation durch B
t_1	pos	pos	neg
t_2	pos	pos	pos
t_3	pos	neg	pos
t_4	pos	pos	neg
t_5	neg	pos	pos
t_6	neg	neg	neg
t_7	neg	pos	neg
t_8	neg	neg	pos

1. Erstellen Sie für beide Klassifikatoren jeweils die Confusion Matrix.
2. Berechnen Sie für beide Klassifikatoren:
   - Precision
   - Recall
   - Accuracy
3. Gegeben seien zwei Klassifikatoren E und F. Sei E_{error} (bzw. F_{error}) die Menge der Beispiele, die E (bzw. F) auf einer Testdatenmenge falsch klassifiziert hat. Der *Komplementaritätswert* von E in Bezug auf F ist wie folgt definiert:

$$Comp(E,F) = \begin{cases} \left(1 - \frac{|E_{error} \cap F_{error}|}{|E_{error}|}\right) \cdot 100, & \text{falls } E_{error} \neq \emptyset \\ 0, & \text{sonst} \end{cases}$$

   a) Berechnen Sie für die in der obigen Tabelle angegebenen Ergebnisse die Werte für $Comp(A,B)$ und $Comp(B,A)$ für die beiden Klassifikatoren A und B.

   b) Erklären Sie mit eigenen Worten, was der Wert $Comp(A,B)$ aussagt.

3 Entscheidungsbäume und die Titanic[1]

In dieser Aufgabe werden Sie mit KNIME einen Entscheidungsbaum zur Vorhersage des ***Überlebensstatus*** von Passagieren der Titanic erstellen. Dazu verwenden Sie einen vorbereiteten, bereinigten Datensatz.

Datensatz herunterladen

Laden Sie die Datei `titanic_cleaned_age.xlsx` von Stud.IP herunter. Der Datensatz enthält verschiedene Informationen zu Passagieren der Titanic, u. a. Alter, Geschlecht, Ticketpreis etc. Ziel ist es, vorherzusagen, ob eine Person überlebt hat (`Survived`).

Daten in KNIME einlesen

- Fügen Sie einen **Excel Reader**-Node in Ihren Workflow ein.
- Wählen Sie die Datei `titanic_cleaned_age.xlsx` aus und lesen Sie sie ein.

Datenvorverarbeitung

Gehen Sie in den Einstellung des **Excel Reader**-Nodes auf *Transformation* und führen Sie die folgenden Schritte aus.

- Löschen Sie die folgenden Spalten, da sie für die Klassifikation nicht hilfreich sind:
  - `PassengerId` (reine Identifikationsnummer)
  - `Name` (zu individuell, keine Verallgemeinerung möglich)
  - `Ticket` (ebenso zu individuell)
  - `Cabin` (viele fehlende Werte, wenig Informationsgehalt)
- Behalten Sie alle übrigen Spalten bei, insbesondere:
  `Pclass, Sex, Age, SibSp, Parch, Fare, Embarked, Survived`
- Stellen Sie sicher, dass die Zielspalte `Survived` vom Typ **String** ist.

Aufteilung in Trainings- und Testdaten

- Verwenden Sie den Node **Partitioning**, um die Daten aufzuteilen.
- Einstellung:
  - Verhältnis: **80% Training**, **20% Testdaten**
  - Methode: **Random** (zufällig)

[1] Aufgabe wurde mit Unterstützung von ChatGPT erstellt.

  - **Random Seed: 42**

- Der obere Ausgang des **Partitioning** Nodes liefert die Trainings- der untere die Testdaten.

Entscheidungsbaum lernen

- Fügen Sie den Node **Decision Tree Learner** ein.
- Verbinden Sie ihn mit dem **oberen Ausgang** des Partitioning-Nodes.
- Konfiguration:
  - Class column: `Survived`
  - Quality measure: `Gain ratio`
  - Pruning method: `No pruning`
  - Haken bei `Reduced Error Pruning` setzen

Visualisierung des Entscheidungsbaums

- Fügen Sie den Node **Decision Tree to Image** hinzu.
- Verbinden Sie ihn mit dem Ausgang des Decision Tree Learners.

Testdaten klassifizieren

- Fügen Sie den Node **Decision Tree Predictor** hinzu.
- Verbinden Sie:
  - unteren Ausgang des **Partitioning**-Nodes mit dem **unteren Eingang**
  - Ausgang des **Decision Tree Learners** mit dem **oberen Eingang**

Modell auswerten (Accuracy, Precision, Recall)

- Fügen Sie den Node **Scorer** hinzu und verbinden Sie ihn mit dem Ausgang des Decision Tree Predictors.
- Konfigurieren Sie:
  - First column: `Survived`
  - Second column: `Prediction(Survived)`
- Führen Sie den Node aus und sehen Sie sich die **Confusion Matrix** an.
- Notieren Sie sich die Precision, Recall und Accuracy.