CART-Klassifikator

Pattern Matching & Machine Learning

F. Freter, E. Kirchberger,
S. Symhoven & J. Wustl

Sommersemester 2023

20. Juni 2022

Hochschule München
University of Applied Sciences
Fakultät für Informatik und Mathematik

Inhalt

Training und Aufbau des Baumes

CART: Classification And Regression Trees

- **Decision Trees**: Graphische Darstellung von Entscheidungen
- **Zweck**: Vorhersage von kategorischen oder kontinuierlichen Zielvariablen
- **Trainingsdaten**: Beispiele mit Merkmalen und Zielvariablen
- **Überwachtes Lernverfahren**: Anwendungen von medizinischer Diagnostik bis Kreditrisikobewertung
- **Verzweigungsknoten**: Auswahl zwischen Alternativen
- **Blattknoten**: Finale Entscheidung (Vorhersage)
- **CART-Algorithmen**: **Klassifizierung** (kategorisch) und **Regression** (kontinuierlich)

CART: Classification And Regression Trees

- **Decision Trees**: Graphische Darstellung von Entscheidungen
- **Zweck**: Vorhersage von kategorischen oder kontinuierlichen Zielvariablen
- **Trainingsdaten**: Beispiele mit Merkmalen und Zielvariablen
- **Überwachtes Lernverfahren**: Anwendungen von medizinischer Diagnostik bis Kreditrisikobewertung
- **Verzweigungsknoten**: Auswahl zwischen Alternativen
- **Blattknoten**: Finale Entscheidung (Vorhersage)
- **CART-Algorithmen**: **Klassifizierung** (kategorisch) und **Regression** (kontinuierlich)

Classification Trees

Im Folgenden fokussieren wir uns auf die **Classification Trees**.

Aufbau eines Classification Trees

- **Root Node**: Startpunkt, enthält alle Daten und startet die Unterteilung (basierend auf Merkmal mit Informationsgewinn).
- **Decision Node**: Teilt Daten weiter auf, basierend auf Merkmalen.
- **Leaf Node**: Endpunkte repräsentieren finale Vorhersagen (basierend auf Merkmalen des gegebenen Datenpunkts). Keine weiteren sinnvollen Teilungen möglich.

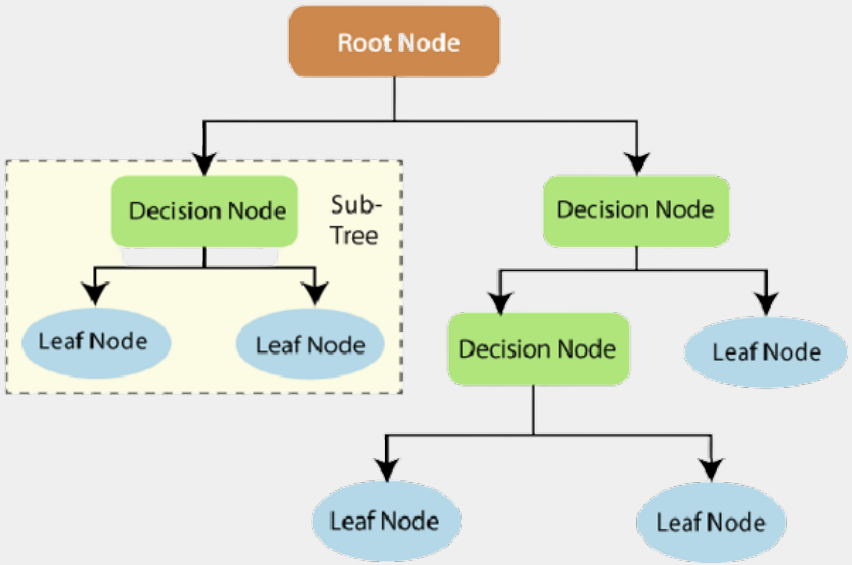

Abbildung: Decision Tree [1]

Aufbau eines Classification Trees

- **Root Node**: Startpunkt, enthält alle Daten und startet die Unterteilung (basierend auf Merkmal mit Informationsgewinn).
- **Decision Node**: Teilt Daten weiter auf, basierend auf Merkmalen.
- **Leaf Node**: Endpunkte repräsentieren finale Vorhersagen (basierend auf Merkmalen des gegebenen Datenpunkts). Keine weiteren sinnvollen Teilungen möglich.

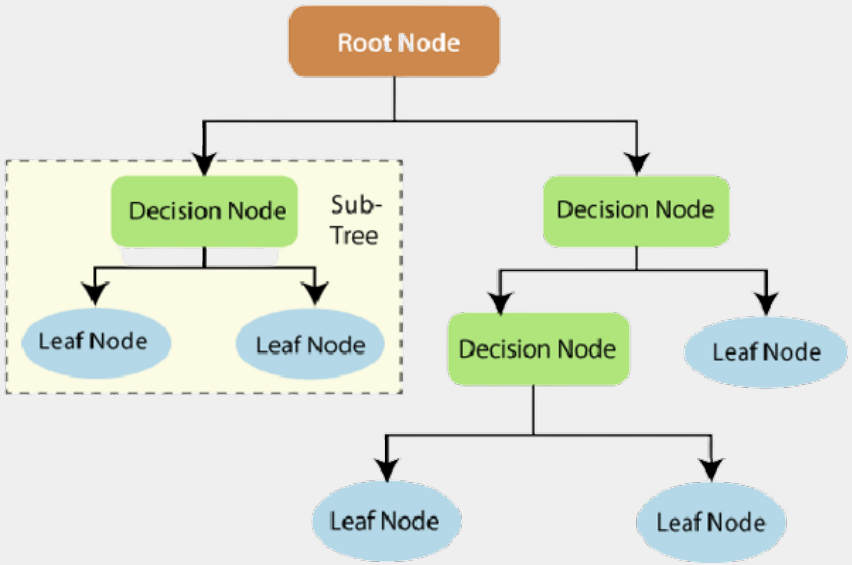

Abbildung: Decision Tree [1]

Ziel

Optimale Vorhersagen auf Basis von Eingangsmerkmalen.

Strategie

- **Allgemeine Strategie**: Eingangsdaten werden in P disjunkte Regionen $R_1, \ldots, R_P$ aufgeteilt.
- Jede Region stellt eine Entscheidungsklasse dar.
- Für jede Region wird die am häufigsten vorkommende Klasse als Vorhersage gewählt.
- Bewertungsmaße wie Gini-Index oder Entropie bestimmen den besten Split.

- **Rekursiver Algorithmus**:
  - Aufteilung des Ausgangsraums R in R_1 und R_2
  - Suche nach der besten Aufteilung für R_1 und R_2
  - Wiederhole für alle erzeugten Regionen

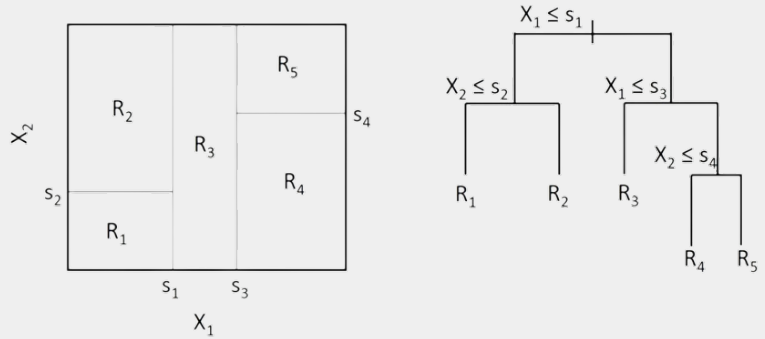

Abbildung: Rekursive Teilung [2]

Vorgehen bei einem Klassifikationsproblem

- Ziel: Vorhersage der Klassenlabel.
- Jeder Knoten repräsentiert eine Entscheidung basierend auf einer Variable.
- Für jede Aufteilungsvariable werden alle möglichen Aufteilungspunkte betrachtet.
- Bewertungsmaße wie Gini-Index oder Informationsgewinn bestimmen die beste Aufteilung.
- Die Aufteilungsvariable und der Aufteilungspunkt, die das optimale Bewertungsmaß erzielen, werden ausgewählt.
- Dieser Prozess wird rekursiv fortgesetzt, bis eine bestimmte Stopp-Regel erfüllt ist (z.B. maximale Tiefe, minimale Anzahl von Instanzen pro Blatt, usw.).
- Jeder Blattknoten repräsentiert eine Klasse; eine neue Beobachtung wird entsprechend klassifiziert.

Bewertungsmasse für einen Split

Bewertungsmasse: Gini-Index, Informationsgewinn & Missclassification Error

- **Gini-Index**
  - Maß der Unreinheit einer Gruppe
  - $Gini = 1 - \sum_{i=1}^{k} p_i^2$, wobei p_i die Wkt. der Klasse i ist.
  - **Ziel**: Minimierung des gewichteten Gini-Indexes.
- **Informationsgewinn**
  - Reduktion der Entropie durch den Split
  - $IG = H(parent) - \sum_{j=1}^{m} \frac{n_j}{n} H(child_j)$, wobei H die Entropie ist.
  - **Ziel**: Maximierung des Informationsgewinns.
- **Missclassification Error**
  - Der Misclassification Error (ME) ist ein Maß für die Fehlklassifizierung.
  - $ME = 1 - \max(p_1, p_2, \ldots, p_k)$, wobei p_i die Wkt. der Klasse i ist.
  - ME kann als Bewertungsmaß für die Baumkonstruktion verwendet werden.
  - **Ziel**: Minimierung des gewichteten Missclassification Errors.

Auswertung des Modells

Auswertung des Modells

- Unvoreingenommene Schätzung der Modellleistung durch Kreuzvalidierung oder separate Testdatensätze.
- Gebräuchliche Metriken für binäre Klassifikation: Genauigkeit, Präzision, Recall, F1-Score und AUC-ROC.
- Genauigkeit: Verhältnis der korrekten Vorhersagen zu den gesamten Vorhersagen.
- Präzision: Verhältnis der wahren Positiven zu der Summe aus wahren und falschen Positiven.
- Recall: Verhältnis der wahren Positiven zu der Summe aus wahren Positiven und falschen Negativen.
- F1-Score: harmonisches Mittel von Präzision und Recall.
- AUC-ROC: Zusammenfassung der Klassifikationsleistung über alle möglichen Klassifikationsschwellen.
- Overfitting-Vermeidung durch Techniken wie Pruning oder Regularisierung.

Overfitting und Pruning

Overfitting in Decision Trees

- Ein vollständig gewachsener Decision Tree kann überangepasst sein (**Overfitting**).
- Dies kann durch Rauschen oder einen Mangel an repräsentativen Daten verursacht werden.
- **Ziel**: Erstellung eines Modells, das gut auf neue, ungesehene Daten verallgemeinert.

Vermeidung von Overfitting

- Das Wachstum des Baums kann vorzeitig gestoppt werden.
- Alternativ kann der Baum zunächst vollständig wachsen und anschließend beschnitten werden (**Pruning**).
- Verschiedene Pruning-Methoden: Reduced Error Pruning, Minimum Description Length Pruning, Cost-Complexity Pruning.

Cost-Complexity Pruning

- **Ziel**: Verhindern von Overfitting durch Entfernen von Zweigen, die wenig zur Vorhersageleistung beitragen
- **Kostenkomplexitätspruning**: Gleichgewicht zwischen Baumgröße und Trainingsfehler
- **Kostenkomplexitätskriterium**:

$$C_\alpha(T) = C(T) + \alpha|T|, \textit{ mit}$$

  - $C(T)$ ist der Misclassification Error des Baumes T.
  - $|T|$ ist die Anzahl der terminalen Knoten des Baumes T.
  - α ist ein Komplexitätsparameter, der die Präferenz zwischen Baumgröße und Trainingsfehler steuert.
- Durch Variieren von α kann eine Sequenz optimaler Bäume ermittelt werden.
- Kreuzvalidierung kann verwendet werden, um den optimalen Wert von α zu bestimmen.

Vor- und Nachteile

Vorteile:

- leicht zu trainieren
- leicht zu interpretieren
- einfach zu visualisieren
- können mit verschiedenen Prädiktoren umgehen
  $\rightarrow$ keine Dummies erforderlich

Nachteile:

- nicht die besten Lerner
- reagieren empfindlich auf sich ändernde Trainingsdaten
- werden von den oben genannten Splits dominiert
  $\rightarrow$ erster Split beeinflusst stark die Form des gesamten Baums

Verbesserungsmöglichkeiten & Ausblick

- **Stacking**: Ensemble-Lern-Technik. Mehrere CART-Modelle kombiniert werden. Ausgaben der einzelnen Modelle werden als Eingabe für ein Meta-Modell verwendet.
- **Bayesian Model Averaging**: Modellselektion. Mehrere Modelle auf der Grundlage von Bayes'schen Wahrscheinlichkeiten kombiniert werden.
- **Bagging**: Ensemble-Lern-Technik. Mehrere CART-Modelle werden auf unterschiedlichen Stichproben der Daten trainiert.
- **Random Forests**: Ensemble-Lern-Modell. Besteht aus vielen unkorrelierten Entscheidungsbäumen, die auf zufälligen Untergruppen der Daten trainiert werden.
- **Boosting**: Ensemble-Lern-Technik. Sequentielle Anordnung von schwachen CART-Modellen, wobei jeder Baum versucht, die Fehler des vorherigen Baums zu korrigieren.

References I

Bahzad Charbuty and Adnan Mohsin Abdulazeez.
Classification based on decision tree algorithm for machine learning.
Journal of Applied Science and Technology Trends, 2021.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman.
The elements of statistical learning, 2009.

Fragen, Kritik oder Anregungen?