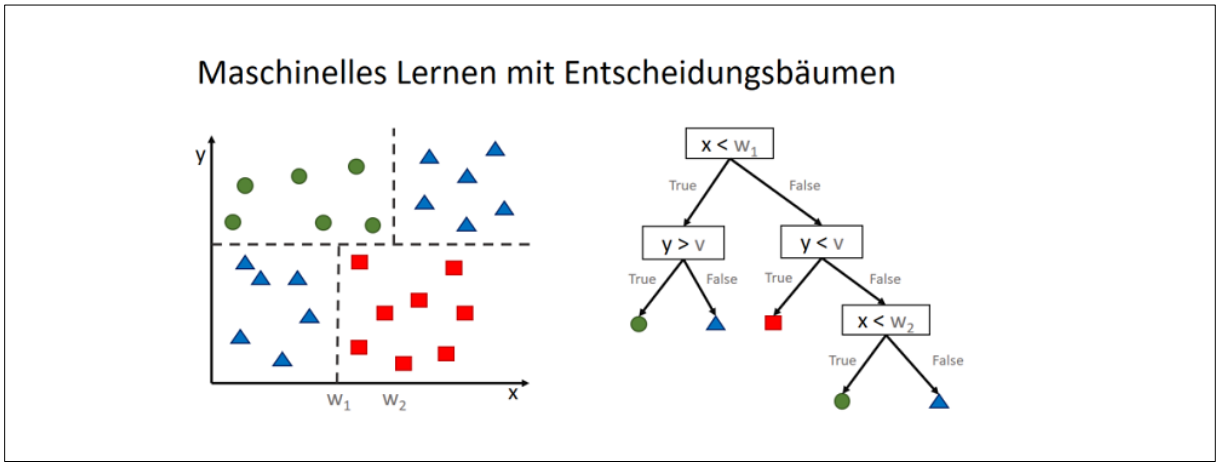
Prof. Dr.-Ing.habil.
Dirk Joachim Lehmann
Data Science in IoT
Fakultät für Informatik
di.lehmann@ostfalia.de
www.dirk-lehmann.de

Kontrafaktische Analyse in Entscheidungsbäumen unter Nebenbedingungen

Hintergrund

Eine traditionelle und etablierte Methode zum Adressieren von Klassifikationsaufgaben dezidierter Datensätze konkreter Domänen sind **Entscheidungsbäume** bzw. deren Erweiterungen wie Entscheidungswälder (Random Forrest). Gegenüber neuronalen Netzen haben Entscheidungsbäume den inhärenten Nachteil lediglich orthogonale Entscheidungsgrenzen zu ermöglichen – wodurch sich nicht beliebige Entscheidungsgrenzen ermöglichen lassen – dem gegenüber jedoch den inhärenten Vorteil durch den Nutzer prinzipiell interpretierbar zu sein. Zusätzlich können komplexe Entscheidungsbäume über Pruning-Ansätze angemessen vereinfacht werden, ohne notwendigerweise die Klassifikationsqualität unangemessen zu verschlechtern. Dadurch werden auch komplexe Entscheidungsbäume durch den Nutzer interpretierbar. In der Praxis wird diese Eigenschaft von Entscheidungsbäume genutzt, z.B. um interpretierbare Surrogate-Modelle von neuronalen Netzen zu erstellen. Derart sollen Black-Box-esque KI-Modelle interpretierbar werden, was unter dem Begriff der „Explainable AI“ (xAI) fällt [4,5,6].



http://datasciencehack.com/wp-content/uploads/2017/02/decision-tree-entscheidungsbaum-verfahren-header.png, Stand: 9.8.2022

Abb.: Ein Entscheidungsbaum klassifiziert für den Nutzer in nachvollziehbarer Art und Weise ein Datenattribut.

In den vergangenen Jahren hat sich mit der **kontrafaktischen Analyse** (*counterfactual analysis*) eine weitere analytische Zielstellung für Klassifikationsmodelle etabliert [1,2,3]. Eine kontrafaktische Analyse fragt nicht mehr nach der konkreten Klassifikation eines Datenattributes, sondern danach, welche Dateneigenschaften angepasst werden müssten, um eine vorgegebene Zielklassifikation zu erreichen. Konkret wird nicht mehr gefragt „Was ist?“, sondern „Was wäre wenn?“. Beispielsweise kann ein Kreditnehmer leider in die Kategorie „nicht kreditwürdig“ fallen. Eine kontrafaktische Analyse fragt nun danach, welche Eigenschaft der Kreditnehmer ändern kann, um in die Kategorie „kreditwürdig“ zu fallen. Beispielweise könnte ein Ergebnis der kontrafaktischen Analyse sein, dass der Kreditnehmer in einen

anderen Stadtteil ziehen muss, um „kreditwürdig“ zu sein, oder er/sie müsste sein/ihr Einkommen um 10% erhöhen, um „kreditwürdig“ zu sein etc.

Eine kontrafaktische Analyse ermittelt welche Dateneigenschaften eine Klassifikation erzeugt bzw. ermittelt einen Datenpunkt in der Nähe des originalen Datenpunktes welcher die gewünschten Klassifikationseigenschaften besitzt.

Kontrafaktische Analysen auf Entscheidungsbäumen sind bisher noch wenig studiert. In diesem Projekt ist daher die Zielstellung ein interaktives System in Python zu entwickelt, welches sowohl Entscheidungsbäume aus geeigneten Daten erzeugt als auch kontrafaktische Analysen ermöglicht.

Kontrafaktische Analyse auf Entscheidungsbäume – Eine Kurzbetrachtung

Gegeben sein ein Entscheidungsbaum E. Ein neues Anfrageelement/Datenattribut **x** gibt die Klassifikation y=E(**x**). Was wir in der kontrafaktischen Analyse möchten ist, für das Anfrageobjekt **x** eine Zielklasse yz vorgeben, als welche Dieses klassifiziert werden soll (obwohl es sonst als Klasse y klassifiziert wird), so dass gilt yz=E(**x**+f(**x**,yz)). f(**x**, yz) ist eine Transferfunktion, welche Attribute/Komponenten in **x** = (x1,....,xn) so abändert, dass die Klasse yz resultiert:

$$f(\mathbf{x}, yz) = \mathbf{dx}$$

mit

$$\mathbf{x} + \mathbf{dx} = \mathbf{x} + f(\mathbf{x}, yz)$$

mit

$$(x1+dx1, x2+dx2, ..., xn+dxn) = (x1, x2, ...xn)+(dx1, dx2, ..., dxn) = \mathbf{x}+\mathbf{dx}$$

mit

E(**x**)=y und E(**x**+f(**x**,yz))=yz.

Was auf die Frage führt - bei gegebener Zielklasse yz - wie ist **dx** zu wählen, damit der Entscheidungsbaum diese Zielklasse yz erzeugt: yz = E(**x**+**dx**). Konkret ist damit die Frage verbunden, wie f(**x**,yz) zu konstruieren ist. Offensichtlich gibt es eine Anzahl unterschiedlicher Versionen der Transferfunktion f, welche auf die gleiche Klasse yz führt. Daher lassen sich unterschiedliche „Wunsch“-Eigenschaften wie **dx** gestaltet sein kann, durch die geeignete Wahl von f(**x**,yz) realisieren. Wir bezeichnen solche Eigenschaften technisch als *Nebenbedingungen* oder *Constraints*.

Beispiele für Nebenbedingungen/Constraints in der Transferfunktion f(**x**,yz):

- (A) **dx** soll so gewählt werden, dass sich lediglich eine minimale Anzahl an Komponenten in **x** ändert.

- (B) ein Maß einer Änderungsmetrik q soll in **dx** einen gewissen Wert c nicht überschreiten:

  q(|| [**x**] – [**x**+f(**x**,yz)]||)=q(||**dx||**) < c

- (C) bestimmte Komponenten in **x** sollen sich nicht verändern dürfen

- …

Solche und weitere Überlegungen sollen in das Projekt mit eingehen.

Angebot

Im Rahmen ihres *Praxisprojektes*, *Masterseminars*, *Masterprojektes*, ihrer *Bachelorarbeit*, *Masterarbeit* oder ähnlichen Studienleistungen - wie z. B. einem *interdisziplinären Digitalisierungsprojekt* - können sie gerne diese Aufgabe bearbeiten

Melden Sie sich gerne bei mir unter: di.lehmenn@ostfalia.de

Aufgaben/Arbeitspakete

Das Projekt besteht aus klar voneinander abgrenzbaren Arbeitspaketen (AP):

- **AP1) Graphical User Interface**
  - Eingabe: Schnittstellen zu allen APs
  - Ausgabe: interaktive User-Interface Systeme um die Parametrisierungen und Monitoring-Aufgaben durchzuführen, Entscheidungsbäume interaktiv zu manipulieren, Nebenbedingung interaktiv zu definieren, etc.

    Beispiele für in diesem Paket umzusetzende Features:
    - Visualisieren von Entscheidungsbäumen
    - Visualisieren von Interaktionen auf Entscheidungsbäumen
    - Ermöglichen von editieren unterschiedlicher gewünschter Nebenbedingungen
    - Widgets für das Ausführen von Actions wie dem Erstellen von Entscheidungsbäumen
    - Visualisieren von Klassifikationspfade im Entscheidungsbaum
    - Edge-Bundling-Konzepte für „dichte“ Entscheidungsbäume
    - …

- **AP2) Entscheidungsbaum-Generator**
  - Eingabe: Daten, Parameter, Entscheidungsbaumalgorithmus
  - Ausgabe: Entscheidungsbaum

    Beispiele für in diesem Paket umzusetzende Features:
    - Realisierung unterschiedlicher Entscheidungsbaum-Algorithmen (4.5, ID3,…)
    - Realisierung unterschiedlicher Pruning-Algorithmen
    - …

- **AP3) Transferfunktion-Generator**
  - Eingabe: Daten, Parameter und Nebenbedingungen, Zielklasse, Entscheidungsbaum
  - Ausgabe: Transferfunktion

    Beispiele für in diesem Paket umzusetzende Features:
    - Realisierung unterschiedlicher Nebenbedingungen (min. A, B, C sind zu realisieren)
    - Berechnung von optimalen Transferfunktionen
    - …

- **AP4) Daten-Accessor**
  - Eingabe: Objekte und Datenadressen
  - Ausgabe: Speicher- und Lade-Operationen

    Beispiele für in diesem Paket umzusetzende Features:
    - Laden/Speichern von Entscheidungsbäumen konkreter Datensätze
    - Laden/Speichern von Transferfunktionen von Entscheidungsbäumen konkreter Datensätze
    - Laden/Speichern von Datensätze in denen kontrafaktische Analysen durchgeführt werden sollen
    - …

Zu den Arbeitspaketen hinzu kommen notwendige Recherchetätigkeiten, Make-Or-Buy-Entscheidungen, Beachtung von Lizenzfragen, Aspekte der Continuous Integration und des Code-Managements, Dokumentationsaufgaben, Fragen zum Aufsetzen/Deployment und der Migration von Entwicklersystemen (und zum Projektmanagement), wie sie in der Softwareentwicklung üblich sind.

Im Rahmen einer Projektbearbeitung wird nicht erwartet, dass unmittelbar alle APs bearbeitet werden können. Je nach Umfang Ihrer zu erbringenden Studierendenleistung können

Teilaspekte einzelner APs bearbeitet werden. Den konkreten, jeweiligen Umfang stimmen wir im Vorfeld gerne gemeinsam ab.

Referenzen

[1] Wachter, Sandra, Brent Mittelstadt, and Chris Russell: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. (2017) https://jolt.law.harvard.edu/assets/articlePDFs/v31/Counterfactual-Explanations-without-Opening-the-Black-Box-Sandra-Wachter-et-al.pdf

[2] Blick in die Blackbox -> Nachvollziehbarkeit von KI-Algorithmen in der Praxis, Bitkom Bundesverband Informationswirtschaft, S23, https://www.bitkom.org/sites/default/files/2019-10/20191016_blick-in-die-blackbox.pdf

[3] If OnlyWe Had Better Counterfactual Explanations:Five Key Deficitsto Rectify in the Evaluation of Counterfactual XAI Techniques, Mark T. Keane,1,2,3Eoin M. Kenny,1,3Eoin Delaney,1,3& Barry Smyth, https://arxiv.org/ftp/arxiv/papers/2103/2103.01035.pdf

[4] Forcing Interpretability for Deep Neural Networksthrough Rule-based Regularization: Nadia Burkart, Philipp M. Faller, Marco F. Huber, https://www.researchgate.net/profile/Marco-Huber/publication/339332634_Forcing_Interpretability_for_Deep_Neural_Networks_through_Rule-Based_Regularization/links/60f3053816f9f313008eb392/Forcing-Interpretability-for-Deep-Neural-Networks-through-Rule-Based-Regularization.pdf

[5] How do Algorithms decide? Peering into the Black Box, https://www.uni-stuttgart.de/en/research/forschung-leben/1-2021/blackbox/

[6] Explanation Framework for Intrusion Detection, Nadia Burkart1, Maximilian Franz1, and Marco F. Huber, https://www.researchgate.net/publication/347925902_Explanation_Framework_for_Intrusion_Detection/fulltext/604cf8b492851c2b23c8fe28/Explanation-Framework-for-Intrusion-Detection.pdf

---

Vorkenntnisse

Es ist hilfreich – aber keine Voraussetzung – wenn Sie Vorkenntnisse/Interesse mitbringen in

- Softwareentwicklung
- Python
- Grundkonzepte des Maschine Learning
- Data Engineering

Vorarbeiten

Kurzeinführung in Python:

https://www.youtube.com/watch?v=x_kYpwi1L1k

OnboardingProzess

Alle wichtigen Zugänge einrichten um die Arbeit am Projekt aufnehmen zu können:

http://46.38.235.241/webpage/dirkfiles/misc/onboarding/OnboardingProzess.pdf

---

Organisatorisches

Projektmanagement per Scrum in Trello:

https://trello.com/b/xvPUBdYe/kontrafaktischeanalyse

Projektcode-Management per GitHub:

https://github.com/DirkJLehmann/KontrafaktischeAnalyse.git

Projektdokumentations-Management:

https://docs.google.com/document/d/1XaP9ZP6JI0ZpnKQNCFtA-4myOBKhHbkhWXh8FyQANMQ/edit?usp=sharing

---

Bei Interesse melden Sie sich bitte unter:

Prof. Dr.-Ing.habil.
Dirk Joachim Lehmann
Data Science in IoT
Fakultät für Informatik
di.lehmann@ostfalia.de