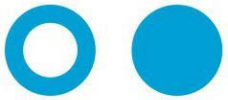


Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Machine Learning zur Klassifikation von Kanten heuristischer TSP-Lösungen

AHMAD KADER

KOLLOQUIUM

FACHBEREICH INFORMATIK

11.07.2024

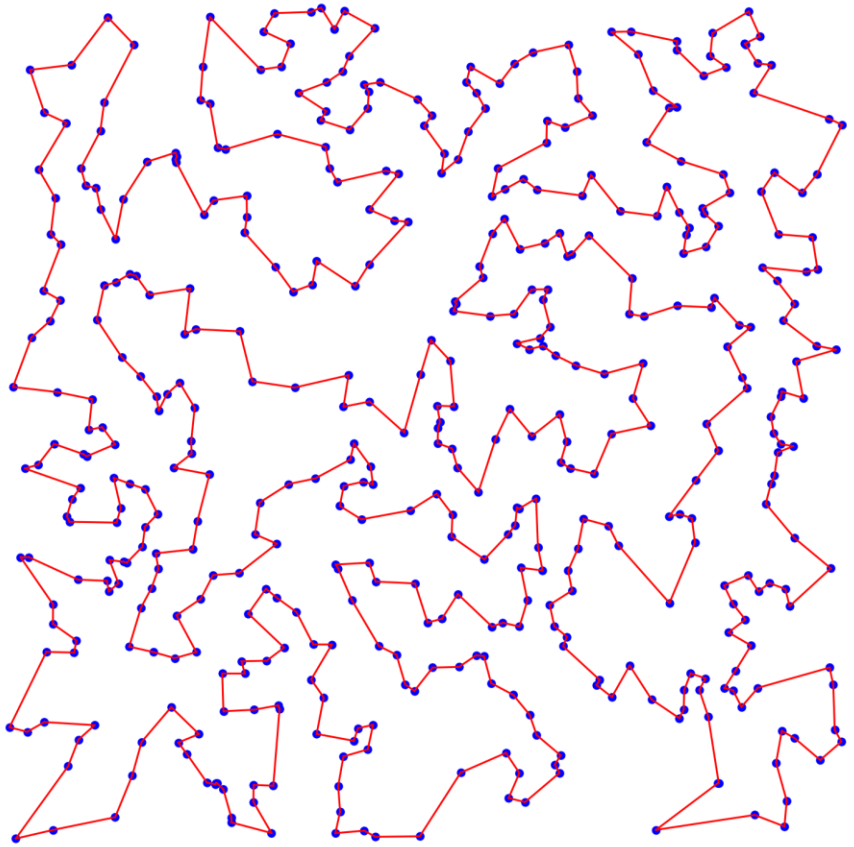
Inhaltsverzeichnis

Gegenstand und Ziel der Arbeit

- Optimierung bestehender Probleme
- Traveling Salesman Problem (TSP)
- Approximationen als Alternative zu der optimalen Lösung
- Erkennung von Mustern
- Ersetzen der Kanten
- Ziel dieser Arbeit:
  - Implementierung von zwei Machine Learning Modellen
  - Implementierung von 20 Merkmalen
  - Vorhersagen treffen, ob eine Kante Teil der optimalen Lösung ist
  - Vorhersagen mit einer Präzision von mindestens 80% treffen

Traveling Salesman Problem

- Graphenproblem $G = (V, E)$
- Knotenmenge $V = \{1, \dots, n\}$
- Kantenmenge $E = \{e_{ij} \mid i, j \in V, i \neq j\}$
- Gewichtungsfunktion $w(e) \rightarrow N_0$ [1]
- Finde eine Tour T:
  - die alle Knoten **genau einmal** besucht
  - im Ursprungsknoten endet
  - die Summe aller Gewichte minimiert



Der Datensatz

- 10000 JSON Dateien
- 1000 Traveling Salesman Probleme
- 1000 optimale Lösungen
- 8000 heuristische Lösungen
- $300 - 500$ Knoten
- Die wichtigsten Attribute:
  - Dimension
  - Koordinaten
  - Tour
  - Tourlänge

Json Dateien	Anzahl
Traveling Salesman Probleme	1000
Optimale Lösungen	1000
Greedy Algorithmus Lösungen	1000
Nearest Neighbor Lösungen	1000
Farthest Insertion Lösungen	1000
Random Insertion Lösungen	1000
Nearest Insertion Lösungen	1000
Cheapest Insertion Lösungen	1000
MST-Heuristik Lösungen	1000
Christofides Algorithmus Lösungen	1000


```
"dimension":477,
"edge_weight_type":EUC_2D,
"name":"fullRandom_large_0",
"node_coordinates":[[82310.0,35609.0],
[40705.0,55707.0],...,[25914.0,82852.0],
[63700.0,60186.0]],
"tour":[0,343,40,...,167,5,450],
"tourlength":1607233,
"type":"TSP"
```

Merkmale

- Implementierung von 20 Merkmale
- 11 Merkmale stammen aus anderen Arbeiten [1] [2]
- 9 Merkmale sind aus dieser Arbeit entstanden
- Alle Merkmale werden aus dem Greedy Algorithmus berechnet

Lokale Merkmale

- $f_1(e_{ij}) = \frac{(1+c_{ij})}{(1+\max\limits_{(l,k)\in E} c_{lk})}$
- $f_2(e_{ij}) = \frac{(1+c_{ij})}{(1+\max\limits_{l\in V} c_{il})}$
- $f_3(e_{ij}) = \frac{(1+c_{ij})}{(1+\max\limits_{l\in V} c_{lj})}$
- $f_4(e_{ij}) = \frac{(1+\min\limits_{(l,k)\in E} c_{lk})}{(1+c_{ij})}$
- $f_5(e_{ij}) = \frac{(1+\min\limits_{l\in V} c_{il})}{(1+c_{ij})}$
- $f_6(e_{ij}) = \frac{(1+\min\limits_{l\in V} c_{lj})}{(1+c_{ij})}$
- f_1 und f_4 berechnen Beziehung von Kante e_{ij} zu allen Kanten des Graphen
- Die anderen berechnen die Beziehung der Kante e_{ij} zu den direkten Nachbarn

Merkmale aus Graphen

- $f_1(e_{ij}) = \dfrac{c_{ij} - \min\limits_{k=1,\dots,n} c_{ik}}{\max\limits_{k=1,\dots,n} c_{ik} - \min\limits_{k=1,\dots,n} c_{ik}}$
- $f_2(e_{ij}) = \dfrac{c_{ij} - \min\limits_{k=1,\dots,n} c_{kj}}{\max\limits_{k=1,\dots,n} c_{kj} - \min\limits_{k=1,\dots,n} c_{kj}}$
- $f_3(e_{ij}) = \dfrac{c_{ij} - \frac{\sum_{k=1}^{n} c_{ik}}{n}}{\max\limits_{k=1,\dots,n} c_{ik} - \min\limits_{k=1,\dots,n} c_{ik}}$
- $f_4(e_{ij}) = \dfrac{c_{ij} - \frac{\sum_{k=1}^{n} c_{kj}}{n}}{\max\limits_{k=1,\dots,n} c_{kj} - \min\limits_{k=1,\dots,n} c_{kj}}$

MST-Merkmal

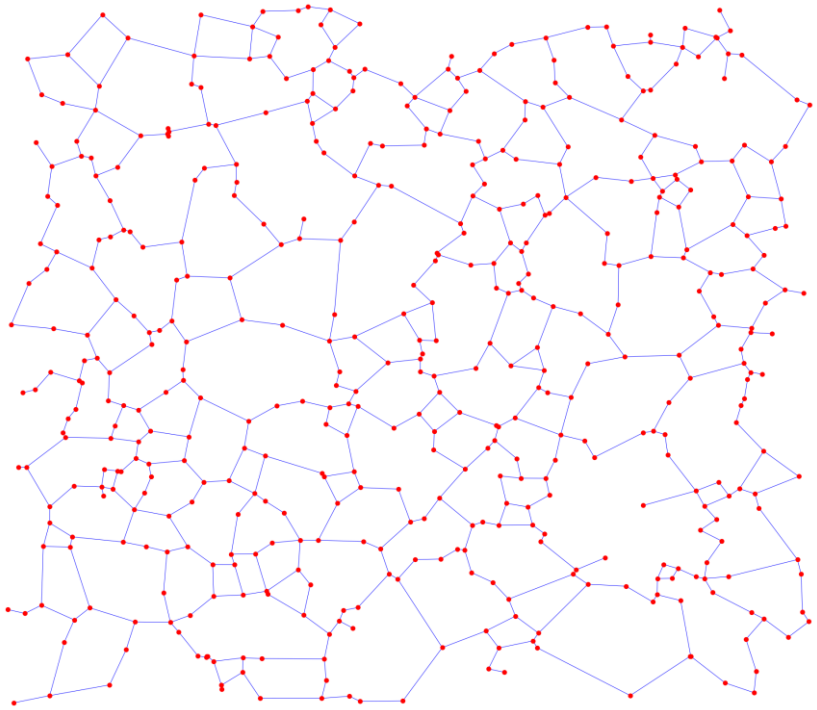
1. Übergebe die heuristischen Tour als Eingabe
2. Wiederhole die nächsten Schritte k-mal
3. Berechne für diese Tour ein MST
4. Entferne die Kanten, die im MST beinhaltet sind
5. Gib den entfernten Kanten eine Gewichtung von k
6. Falls noch Kanten übrig geblieben sind gib ihnen die Gewichtung $2 * k$

- Die Lösung eines MSTs hat eine große Beschneidung zu den Lösungen der optimalen Lösung
- Dient deshalb auch gut als eine untere Schranke

k-RNG-Merkmal

- k-relative Neighborhood Graph
- k-RNG ist ein Graph $G = (V, E)$
- Knotenmenge $V = \{1, \dots, n\}$
- Kantenmenge $E = \{e_{ij} \mid d(u,i) < d(i,j)\ \&\ d(u,j) < d(i,j)\}$
- Es gibt $k - 1$ Knoten u die näher sind an zwei Knoten i, j
- i, j sind relative Nachbarn, wenn diese Bedingung gilt

k	Absolute Zahlen	Prozentanteile
1	328339	81,87
2	54893	13,69
3	13051	3,25
4	3407	0,85
5	974	0,24
6	288	0,07
7	80	0,02
8	29	0,01
9	8	0
10	1	0
11	0	0
12	1	0



Heuristische Merkmale

- Je ein Merkmal für je eine Heuristik
- Überprüfe für jede Heuristik, ob $e_{ij} \in optimal$
- Speichere eine 1 falls enthalten und 0 falls nicht
- Summe der Kanten der Heuristiken
- Berechne für jede Kante e_{ij} die Summe der Kanten, die in der optimalen Lösung sind

Klassifikation

- Welche Kanten vom Greedy Algorithmus sind in der optimalen Lösung
- Abgespeichert in einer binären Liste
- Machine Learning Modelle:
  - Random Forest Tree
  - XGBoost
- Daten werden getrennt:
  - 80% Trainingsdaten
  - 20% Testdaten
- Hyperparametrisierung
  - Anzahl der Entscheidungsbäume
  - Tiefe der Entscheidungsbäume
  - Lernrate

Evaluation

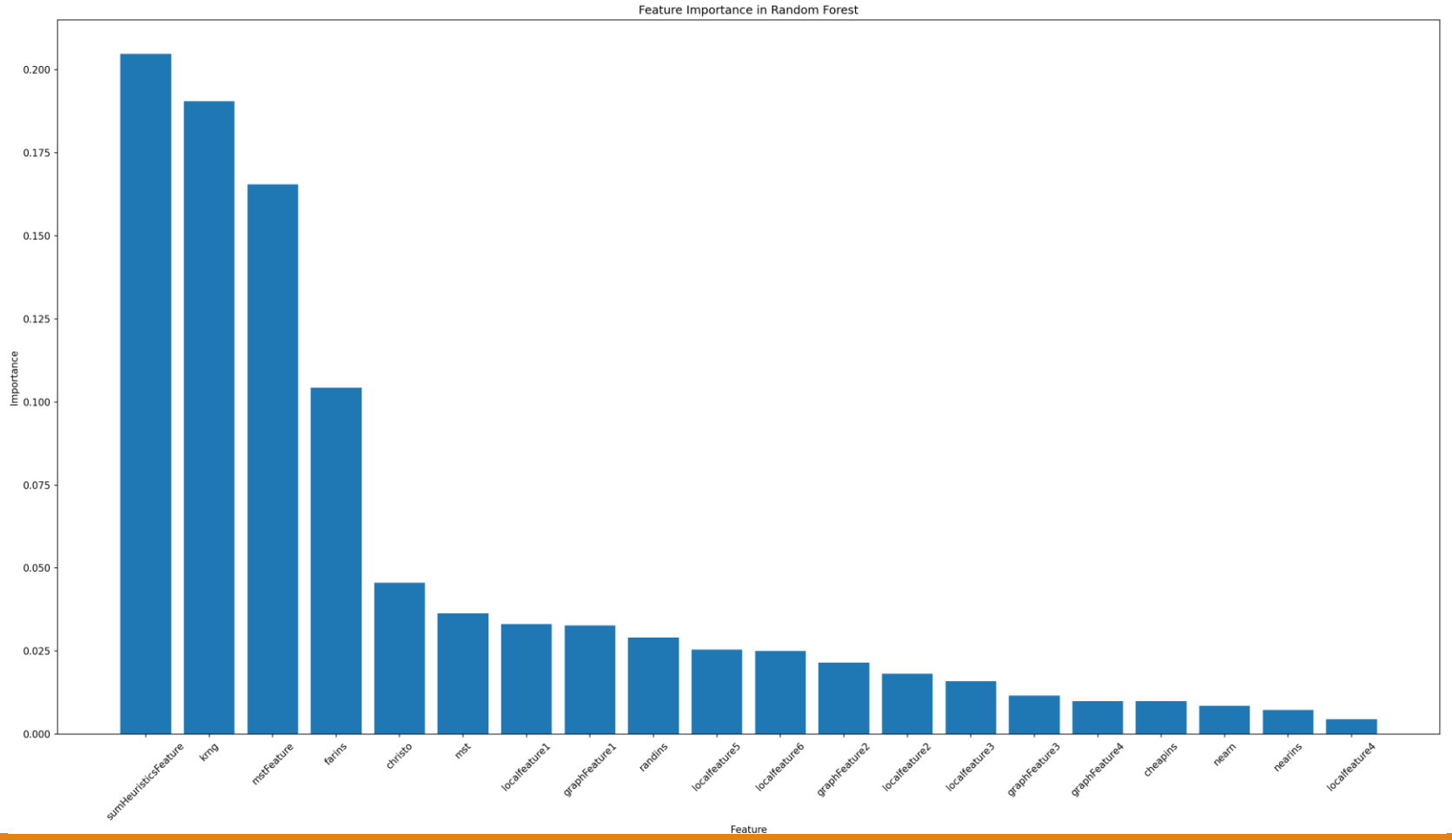
- Konfusionsmatrix für je Machine Learning Modell
- Feature Importance für die Wichtigkeit der Merkmale
- Wichtigste Kennzahlen [3]:
  - $Precision = \frac{|true_{positives}|}{|true_{positives}| + |false_{positives}|}$
  - $Recall = \frac{|true_{positives}|}{|true_{positives}| + |false_{negatives}|}$
  - $F1 - Score = \frac{Precision * Recall}{Precision + Recall}$
- Daten werden zufällig getrennt:
  - 80% Trainingsdaten
  - 20% Testdaten

Random Forest Tree

- ca. 84,5% der Kanten sind richtig positiv
- ca. 78,5% der Kanten sind falsch negativ
- Precision von ca. 84,5%
- Recall von ca. 84,8%
- F1-Score von ca. 84,6%

		Vorhergesagt	
		nicht enthalten	enthalten
Wahrheit	nicht enthalten	10213	10414
	enthalten	2795	56793

Feature Importance für das RFT

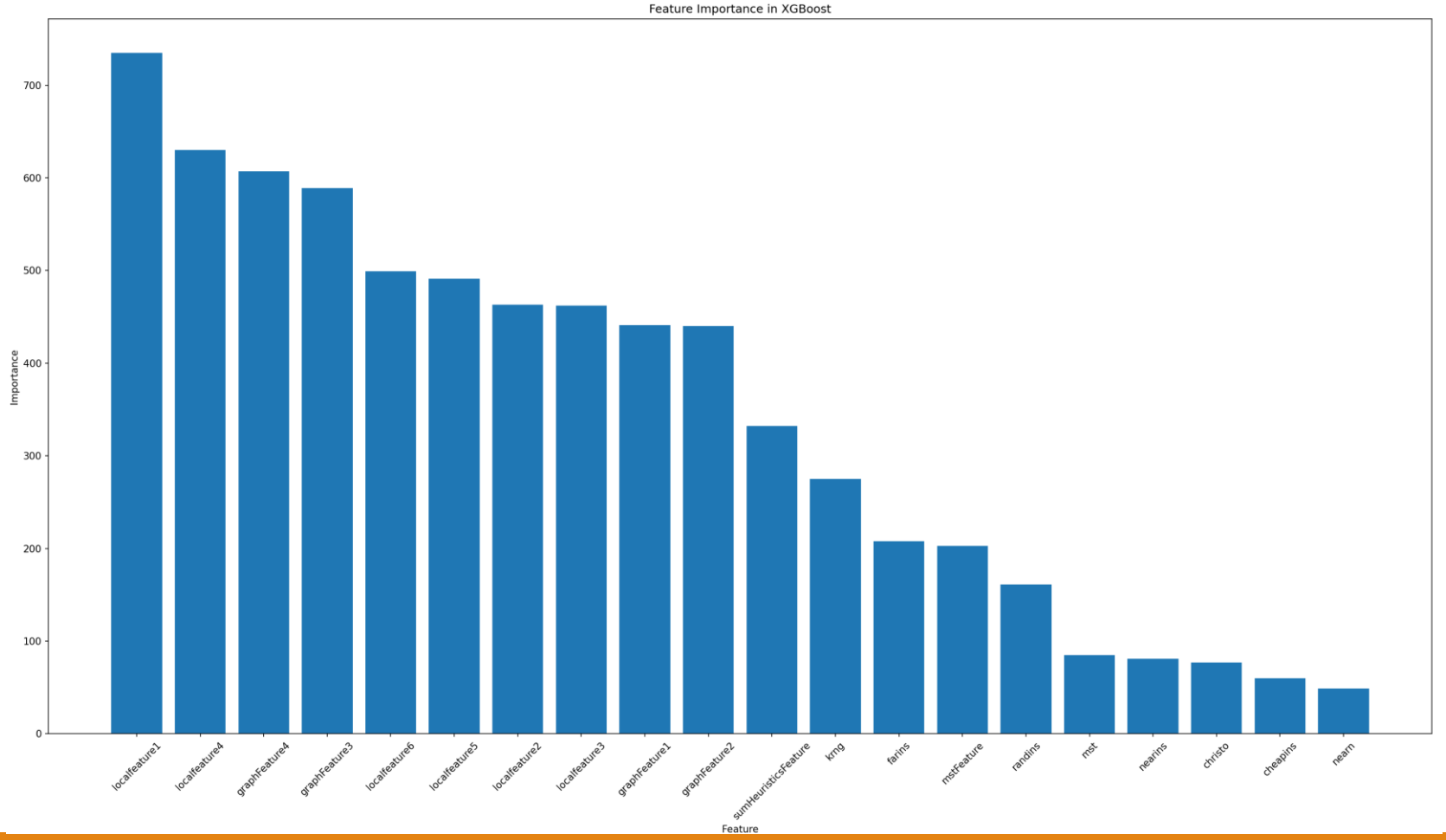


XGBoost

- ca. 84,8% der Kanten sind richtig positiv
- ca. 76,7% der Kanten sind falsch negativ
- Precision von ca. 84,8%
- Recall von ca. 84%
- F1-Score von ca. 84,4%

Wahrheit	Vorhergesagt: nicht enthalten	Vorhergesagt: enthalten
nicht enthalten	10695	10073
enthalten	3240	56207

Feature Importance für das XGBoost



Schlussfolgerung

- Heuristische Lösungen können Klassifiziert werden
- Die Machine Learning Modelle liefern vergleichbare Lösungen
- Die vorgestellten Modelle liefern Vorhersagen mit einer Präzision von mindestens 80%
- Dennoch können beide Modelle noch verbessert werden
- Verbesserung der Lösung durch:
  - Nutzung verschiedener Modelle (z.B. Support Vector Machines oder lineare Regression)
  - Nutzung weiterer Daten
  - Implementierung weiterer Merkmale
  - Klassifikation anderer heuristischer Lösungen

Zusammenfassung

- Problemstellung und das Ziel der Arbeit erklärt
- Traveling Salesman Problem zusammengefasst
- Den Datensatz vorgestellt
- Die implementierten Merkmale erklärt
- Den Verlauf der Klassifikation aufgezeigt
- Ergebnisse vorgestellt und evaluiert
- Eine Schlussfolgerung gezogen

Literaturverzeichnis

- [1] J. Fitzpatrick, D. Ajwani, and P. Carroll, “Learning to sparsify travelling salesman problem instances,” CPAIOR, pp. 410–426, 2021.
- [2] Y. Sun, A. Ernst, X. Li, and J. Weiner, “Generalization of machine learning for problem reduction: a case study on travelling salesman problems,” OR Spectrum, vol. 43, pp. 607–633, sep 2020.
- [3] L. Derczynski, “Complementarity, F-score, and NLP evaluation,” in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16) (N. Calzolari, ed.), (Portoroˇz, Slovenia), pp. 261–266, European Language Resources Association (ELRA), may 2016.