K-Nearest-Neighbour Algorithmus

Šerif Hamzić
München, 05.12.2022

Agenda

- Der Algorithmus
- Stärken
- Schwächen
- Unterschiedliche Implementierungen
- Ergebnisse auf den Testdaten
- Diskussion

Der Algorithmus

Der Algorithmus

- “Du bist mit wem du dich umgibst.”
- Entwickelt 1951 von Evelyn Fix und Joseph Hodges
- Ein Klassifikationsalgorithmus
- Kann auch für Regression verwendet werden
- “Lazy Learning” Algorithmus

KNN als Klassifikationsalgorithmus

- Gegeben:
  - Ein Trainingsdatensatz T mit N Instanzen
  - Ein Hyperparameter k → 1 <= k <= | N |
  - Eine Distanzfunktion f
  - Eine Testinstanz t

- Finde die k ähnlichsten (in Bezug auf f) Instanzen zu t
- Schaue dir ihre Klassen an
- Weise t die Klasse zu, welche am häugisten vorkommt

KNN für Regression

- Gegeben (wie vorher):
  - Ein Trainingsdatensatz T mit N Instanzen
  - Ein Hyperparameter k → 1 <= k <= | N |
  - Eine Distanzfunktion f
  - Eine Testinstanz t

- Finde die k ähnlichsten (in Bezug auf f) Instanzen zu t
- Schaue dir ihre Werte für das Attribut a an
- Das Attribut a von t ist der Mittelwert

“Lazy Learning”

- KNN ist ein “Lazy Learning” Algorithmus
- “Lazy Learning” → Klasse von ML Lernverfahren
- Modell wird nicht zur Zeit des Trainings gebildet, sondern zur Zeit der Anfrage
- Bei KNN besteht das “Modell” einfach aus den abgespeicherten Trainingsinstanzen

Hyperparameter k

- Liegt zwischen 1 <= k <= | N |
- Oft ungerade Zahl
- K = 1 → Nearest Neighbour Search
- K = | N | → Häufigste Klasse im Trainingsdatensatz gewinnt
- Ist k zu klein → Klassifizierer empfindlich gegen Rauschen
- Ist k zu groß → nicht relevante Instanzen fließen in die Entscheidung mit ein; Grenzen zwischen Klassen werden unscharf
- Mögliche Lösung: Gewichtung der Abstände

Gewichtung der Abständen

- Uniforme Gewichtung → alle k Nachbarklassen haben gleiches Gewicht, unabhängig von der Distanz des Nachbars zur Testinstanz t
- Schlecht bei ungleich verteilten Daten / wenigen Punkten
- Stattdessen:
  - Klassen von Instanzen die näher zur Testinstanz t sind, kriegen ein höheres Gewicht

Abstandsfunktion f

Abhängig davon was die zu klassifizierenden Instanzen sind und welche Werte ihre Attribute annehmen können.

- Hier nur ein einige davon:
  - Euklidischer Abstand
  - Manhattan-Distan z
  - Kosinusähnlichkeit
  - Hamming-Distanz

Stärken

Stärken

- Einfach zu verstehen (intuitiv)
- Relativ schnell zu implementieren
- Daten müssen nicht linear trennbar sein
- Klassifikationsentscheidung ist leicht interpretierbar

Schwächen

Schwächen

- Teuer → Hoher Rechen- und Speicheraufwand
- Langsam → Naive Implementierung hat Zeitkomplexität $O(dN^2)$
- Eignet sich nicht für Latenzempfindliche Anwendungen
- Wahl der Features hat großen Einfluss auf die Leistung des Algorithmus → Feature Engineering

Implementierungen

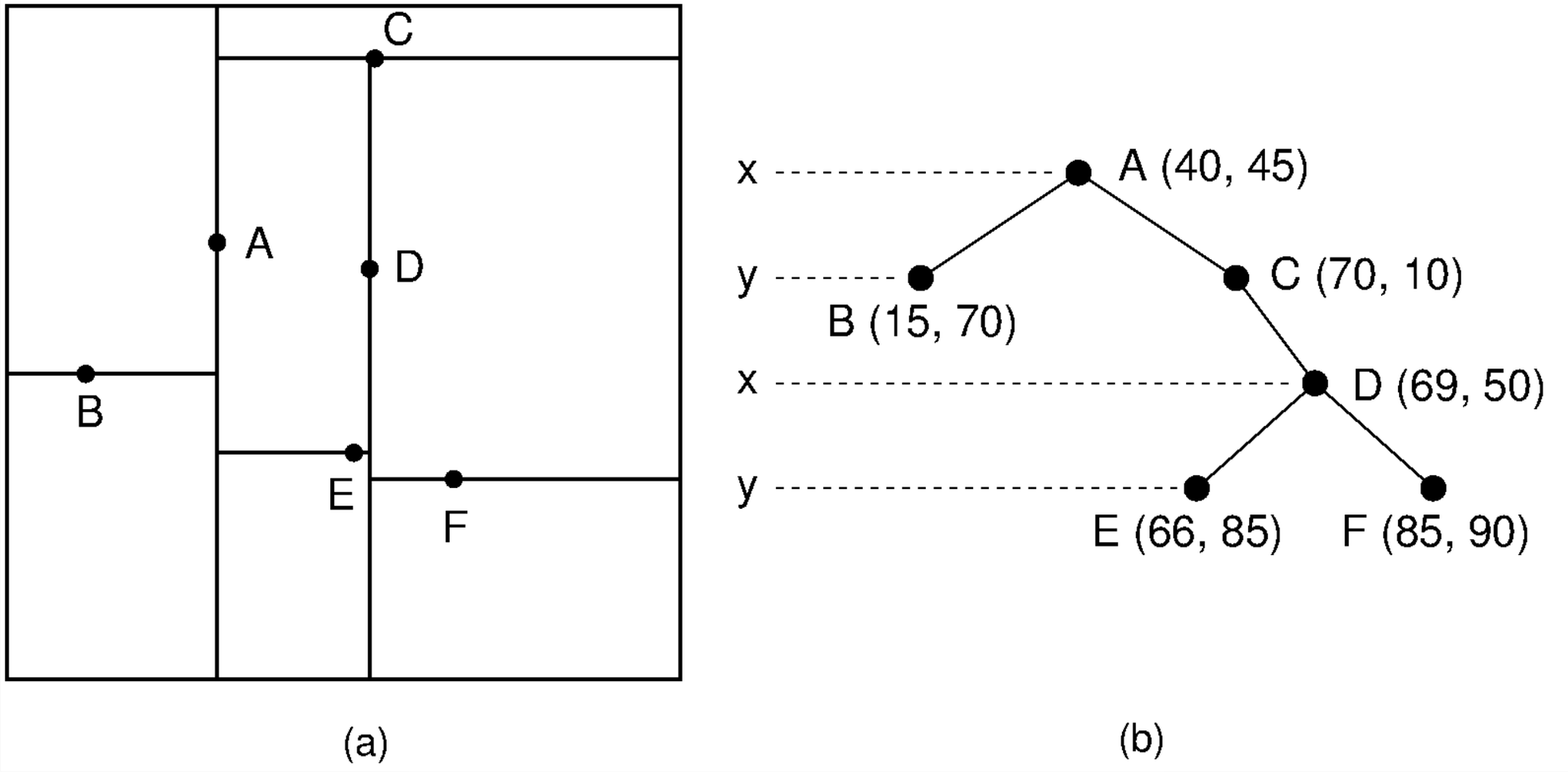
Brute Force Implementierung

- Berechne für jede Testinstanz t die Ähnlichkeit zu allen Trainingsinstanzen und merke dir dabei immer die k nähesten Instanzen.
- Briefsammlung:
  - 39077 Trainingsinstanzen
  - 4881 Testinstanzen
  - 4881 x 39077 die Distanz berechnen
  - 190 734 837 Berechnungen + sortieren
- Laufzeitkomplexität : $O(dN^2)$
- Effektiv, aber schon bei etwas größeren Datenmengen zu ineffizient!

Kd-Baum Implementierung

- Viele der Trainingsbeispiele können im Voraus ausgeschlossen werden → weniger Berechnungen
- Anstatt die Trainingsbeispiele in einem Array abzuspeichern, werden sie in einen Binärbaum eingetragen → kd-Baum

Kd-Baum Implementierung



Kd-Baum Implementierung

- Finden des nearest neighbour ($k=1$) → $O(\log(N))$
- Finden von k-nearest neighbour → $O(DN\log(N))$

- Weitere Implementierungen
  - Ball Tree
  - Best bin first
  - ...

Ergebnisse auf den Testdaten

Briefsammlung - Authoren

TARGET / OUTPUT	Busch	Ibsen	Joyce	Kafka	Goethe	Schiller	Wolf	SUM
Busch	510 10.45%	1 0.02%	0 0.00%	6 0.12%	3 0.06%	3 0.06%	0 0.00%	523 97.51% 2.49%
Ibsen	3 0.06%	880 18.03%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	883 99.66% 0.34%
Joyce	0 0.00%	5 0.10%	557 11.41%	0 0.00%	0 0.00%	0 0.00%	32 0.66%	594 93.77% 6.23%
Kafka	68 1.39%	6 0.12%	0 0.00%	259 5.31%	21 0.43%	30 0.61%	0 0.00%	384 67.45% 32.55%
Goethe	23 0.47%	2 0.04%	0 0.00%	5 0.10%	143 2.93%	57 1.17%	0 0.00%	230 62.17% 37.83%
Schiller	23 0.47%	3 0.06%	0 0.00%	10 0.20%	61 1.25%	176 3.61%	0 0.00%	273 64.47% 35.53%
Wolf	0 0.00%	0 0.00%	125 2.56%	0 0.00%	0 0.00%	0 0.00%	1869 38.29%	1994 93.73% 6.27%
SUM	627 81.34% 18.66%	897 98.10% 1.90%	682 81.67% 18.33%	280 92.50% 7.50%	228 62.72% 37.28%	266 66.17% 33.83%	1901 98.32% 1.68%	4394 / 4881 90.02% 9.98%

Briefsammlung - Authoren

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Busch	0.98	0.02	0.81	0.19	0.89
Ibsen	1.00	0.00	0.98	0.02	0.99
Joyce	0.94	0.06	0.82	0.18	0.87
Kafka	0.67	0.33	0.93	0.07	0.78
Goethe	0.62	0.38	0.63	0.37	0.62
Schiller	0.64	0.36	0.66	0.34	0.65
Wolf	0.94	0.06	0.98	0.02	0.96
Accuracy	0.90				
Misclassification Rate	0.10				
Macro-F1	0.82				
Weighted-F1	0.90				

Briefsammlung - Sprachen

TARGET / OUTPUT	de	unknown	da	fr	it	en	SUM
de	1442 29.54%	4 0.08%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	1446 99.72% 0.28%
unknown	0 0.00%	2 0.04%	1 0.02%	0 0.00%	0 0.00%	0 0.00%	3 66.67% 33.33%
da	1 0.02%	6 0.12%	837 17.15%	0 0.00%	0 0.00%	0 0.00%	844 99.17% 0.83%
fr	0 0.00%	0 0.00%	0 0.00%	36 0.74%	0 0.00%	3 0.06%	39 92.31% 7.69%
it	0 0.00%	1 0.02%	0 0.00%	0 0.00%	30 0.61%	4 0.08%	35 85.71% 14.29%
en	0 0.00%	3 0.06%	0 0.00%	0 0.00%	1 0.02%	2510 51.42%	2514 99.84% 0.16%
SUM	1443 99.93% 0.07%	16 12.50% 87.50%	838 99.88% 0.12%	36 100.00% 0.00%	31 96.77% 3.23%	2517 99.72% 0.28%	4857 / 4881 99.51% 0.49%

Briefsammlung - Sprachen

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
de	0.997	0.003	0.999	0.001	0.998
unknown	0.667	0.333	0.125	0.875	0.211
da	0.992	0.008	0.999	0.001	0.995
fr	0.923	0.077	1.000	0.000	0.960
it	0.857	0.143	0.968	0.032	0.909
en	0.998	0.002	0.997	0.003	0.998
Accuracy	0.995				
Misclassification Rate	0.005				
Macro-F1	0.845				
Weighted-F1	0.994				

Huffington Post - Kategorien

- 34 Kategorien
- Sehr schlecht!
- Macro-Averaged F1 = 5.4 %

Sentiment Daten

TARGET / OUTPUT	positive	negative	SUM
positive	377 37.66%	120 11.99%	497 75.86% 24.14%
negative	107 10.69%	397 39.66%	504 78.77% 21.23%
SUM	484 77.89% 22.11%	517 76.79% 23.21%	774 / 1001 77.32% 22.68%

Sentiment Daten

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
positive	0.759	0.241	0.779	0.221	0.769
negative	0.788	0.212	0.768	0.232	0.778
Accuracy	0.773				
Misclassification Rate	0.227				
Macro-F1	0.773				
Weighted-F1	0.773				

Diskussion