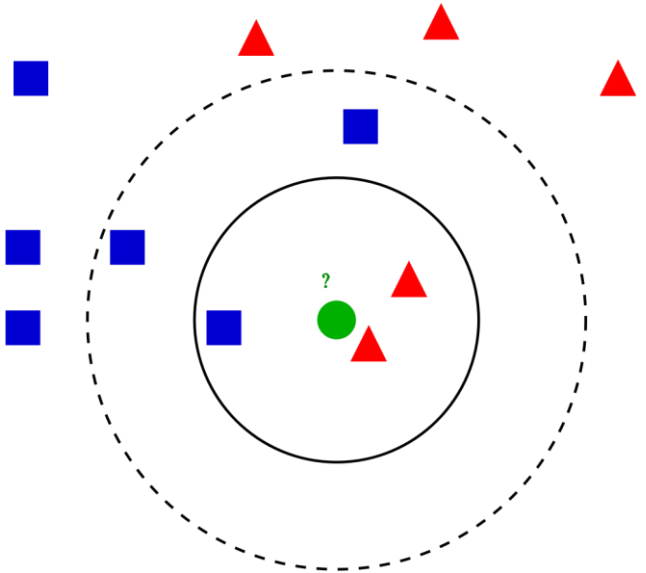
Reverse k Nearest Neighbour Query

Christian Brzeski

k Nearest Neighbour

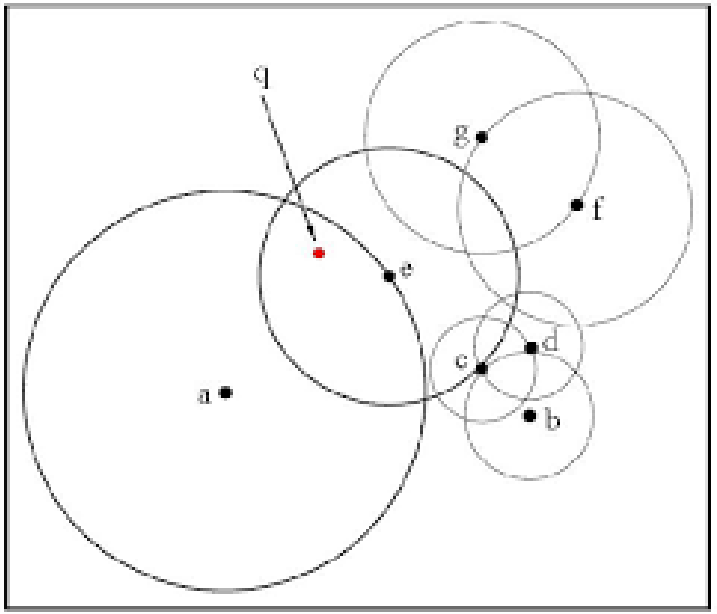
- Methode, welche für Klassifikation und Regression genutzt wird
- Input: k nächste Trainingseinheiten
- Output: Zugehörige Klasse
- Trainingseinheiten sind Feature Vektoren in einem multidimensionalen (meist) euklidischem Raum



Quelle: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Reverse kNN

- Beim Reverse kNN Verfahren wird das Prinzip umgekehrt
- Für einen Query- Vektor q werden alle Datenpunkte gesucht, für die q ein Nearest Neighbour ist



Quelle: http://www.aamircheema.com/thesis/node22.html

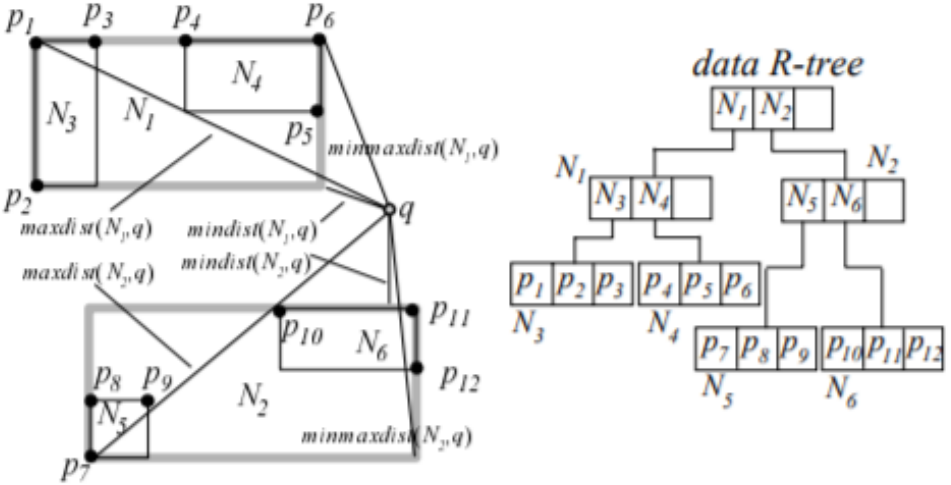
Definition

Gegeben einer Menge an Objekten P und einem Query- Objekt q ist das Ziel einer Reverse kNN- Query eine Menge an Objekten RNN zu finden, sodass für jedes Objekt $p \in P$ und r $\in RNN$

$dist(q,\ r) \leq dist(p,\ r)$ gilt.

Annahmen

- Datenmenge wird mittels R-Baum indiziert
- RNN- Algorithmen bestehen aus zwei Schritten:
  - Filterung – Einige Datenpunkte aus P, um Regionen zu filtern, wo keine NN vorkommen
  - Verifikation – Überprüfung, ob es sich bei den restlichen p um NN handelt



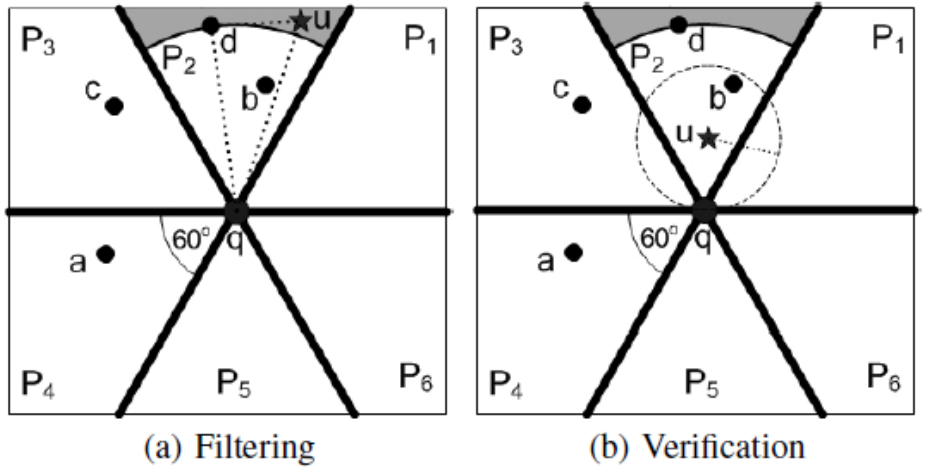
Quelle: Y. Tao, et.al. Reverse kNN Search in Arbitrary Dimensionality

Six-Regions Algorithmus

- Um Query- Objekt q werden 6 Regionen (60°) gebildet
- Für jede Region werden die k Nearest Neighbour von q ermittelt (Filter)
- Für jeden Datenpunkt (a - d) werden die k nächsten Nachbarn ermittelt (Verifizierung)

\+ Gut geeignet für zweidimensionalen Raum

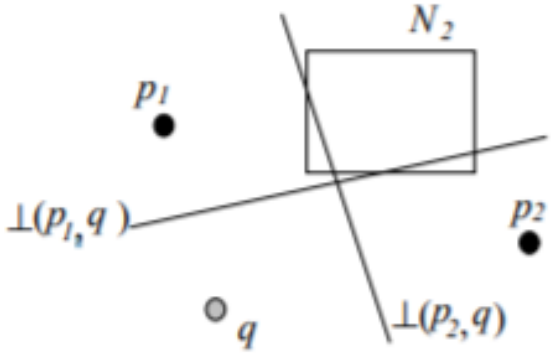
\- Schlechte Performance für höhere Dimensionen



Quelle: S. Yang et.al. Reverse k Nearest Neighbors Query Processing

TPL Algorithmus

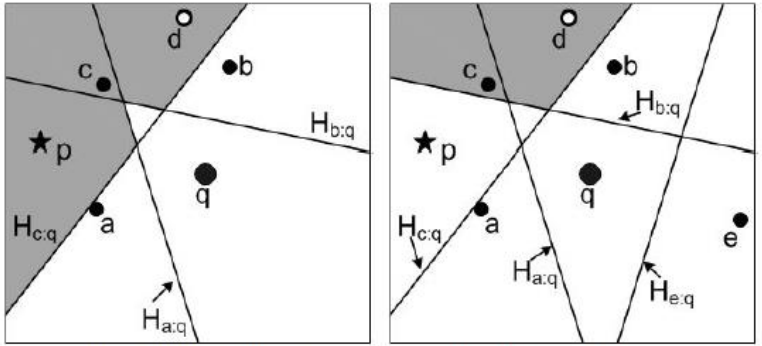
- Objekte werden nach Entfernung zu q in einem Heap sortiert und der Reihe nach untersucht
- Raum wird zwischen den einzelnen Punkte p_i und q in der Mitte durch $H_{p_i:q}$ geteilt
- Wird ein Punkt durch $\sum_i H_{i:q} \geq k$ von q getrennt, dann wird dieser gefiltert



Quelle: Y. Tao, et al. Reverse kNN Search in Arbitrary Dimensionality

TPL Algorithmus

- Filtermenge in a) besteht aus $S_{fil} = \{a, b, c\}$ mit $\{a, b\}, \{b, c\}, \{c, a\}$
- $\binom{m}{k}$ Menge der zu Filternden Halbräume



Quelle: S. Yang et.al. Reverse k Nearest Neighbours Query Processing

- Filtermenge in b) besteht aus $S_{fil} = \{a, b, c, e\}$ mit $\{a, b\}, \{b, c\}, \{c, e\}, \{e, a\}$
- Beim relaxierten Verfahren reduziert sich die Menge, der zu filternden Halbräume auf m

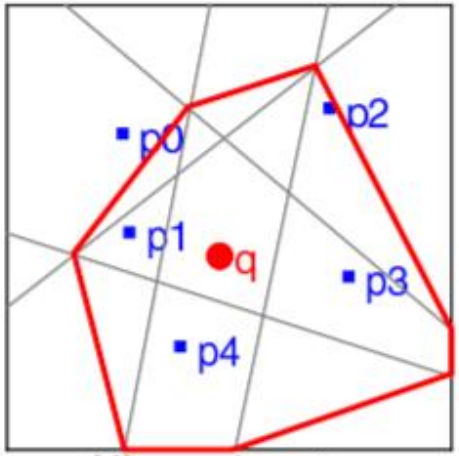
TPL Algorithmus

Verifikation:

- Rechenaufwand für Filterung von Objekt: $O(km)$
- Punkte die nicht gefiltert werden, verbleiben in der Menge S_{cnd} mit potentiellen RKNN
- Nearest Neighbour werden für jeden Kandidaten überprüft
- Falls unter den k Nearest Neighbour q nicht dabei ist, werden diese aus der Menge entfernt

(F)INCH Algorithmus

- Konvexes Polygon liefert Kandidaten
- Rechenaufwand zur Filterung eines Objektes beträgt $O(\log m)$
- Lediglich eine Filtermenge benötigt statt $\binom{m}{k}$

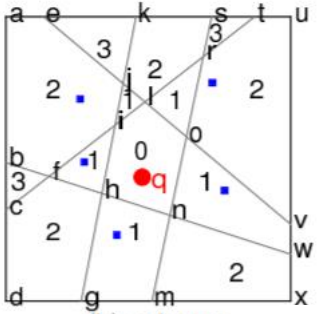


(d) search region

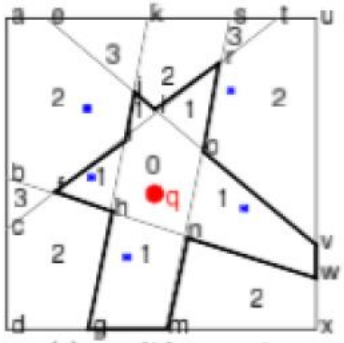
(F)INCH Algorithmus

INCH Algorithmus:

1) Berechne Halbräume für jedes $p \epsilon S_{fil}$
2) Berechne die Schnittpunkte für Halbräume $< k$
3) Berechne das Level jedes Schnittpunkts
4) I:= Die Menge aller Schnittpunkte mit einem Level $< k$
5) Berechnung des Polygons mit Eckpunkten in Menge I

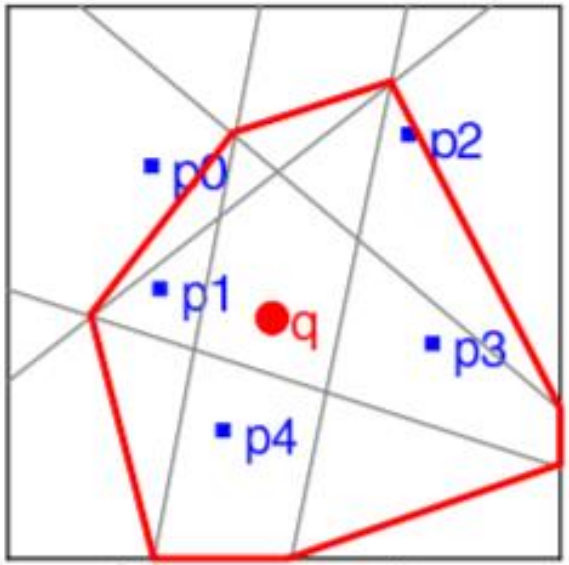


(b) polygons



(c) candidate region

(F)INCH Algorithmus

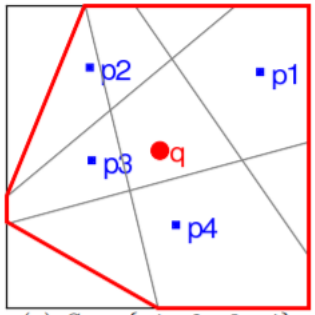


(d) search region

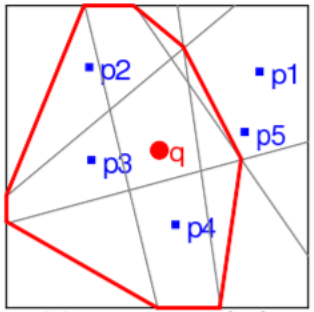
(F)INCH Algorithmus

FINCH Algorithmus:

1) $S_{fil} = \emptyset$
2) $S_{cnd} = \emptyset$
3) *while object* $p \in P - S_{fil}$ *can be found in* S_{cnd} *do*
   - $S_{fil} \coloneqq S_{fil} \cup p$
   - $S_{cnd} \coloneqq INCH(q, k, S_{fil})$
4) *Return* S_{fil}



(a) $S_1 = \{p1, p2, p3, p4\}$



(b) $S_2 = S_1 \cup \{p5\}$

Zusammenfassung

Zusammenfassung:

- Anwendung von Reverse kNN in Marketing, Logistik, Computer Vision
- Bei großen Datenmengen ist eine Indizierung mittels R-Baum sinnvoll
- Performance stark abhängig von der Filterung der einzelnen Algorithmen
- FINCH Algorithmus dominiert andere Verfahren erst ab großen Datenmengen

Quellverzeichnis

- S. Yang, M. A. Cheema, X. Lin, W. Wang, Reverse k Nearest Neighbours Query Processing: Experiments and Analysis, *Proceedings of the VLDB Endowment*, Vol. 8, No. 5, 2015
- F. Korn, S. Muthukrishnan, Influence Sets based on Reverse Nearest, 2000
- Y. Tao, D. Papadias, X. Lian, Reverse kNN Search in Arbitrary Dimensionality, *Proceedings of the 30th VLDB Conference*, 2004
- W. Yang, W. Wu, C. Chang, K. Tan, FINCH: Evaluating Reverse k-Nearest-Neighbour Queries on Location Data, *VLDB Endowment,* 2008
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm