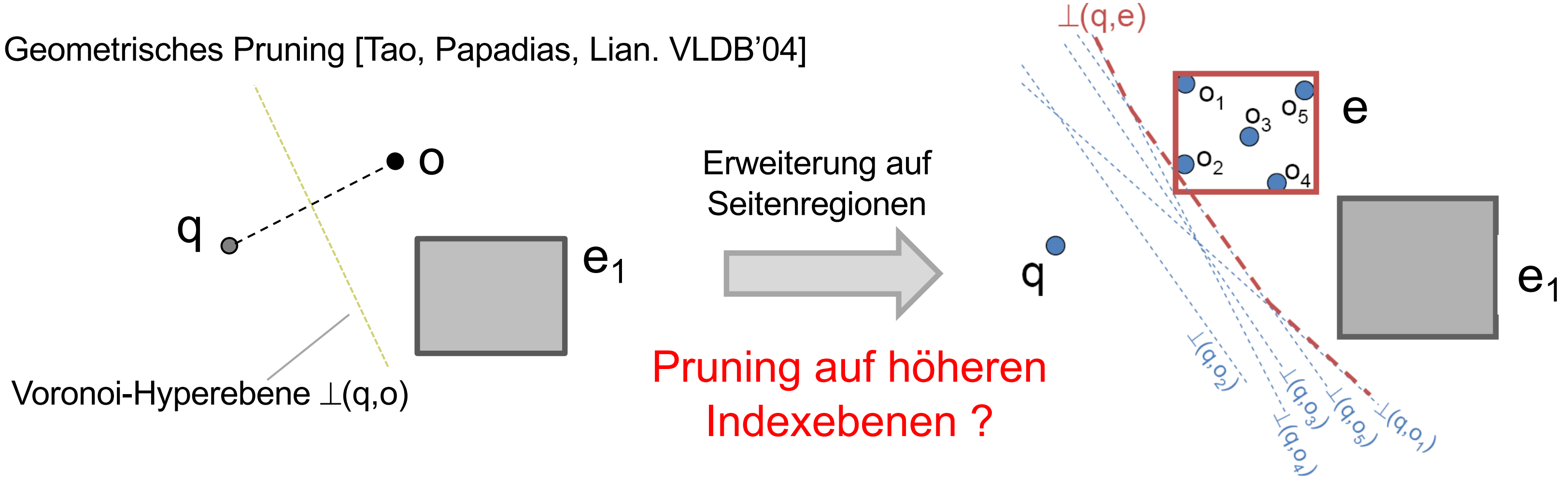
2.5 Reverse nächste Nachbarn Anfragen

- Erweiterung des geometrischen Pruning-Konzepts auf Basis von Voronoi-Hyperebenen auf Seitenregionen

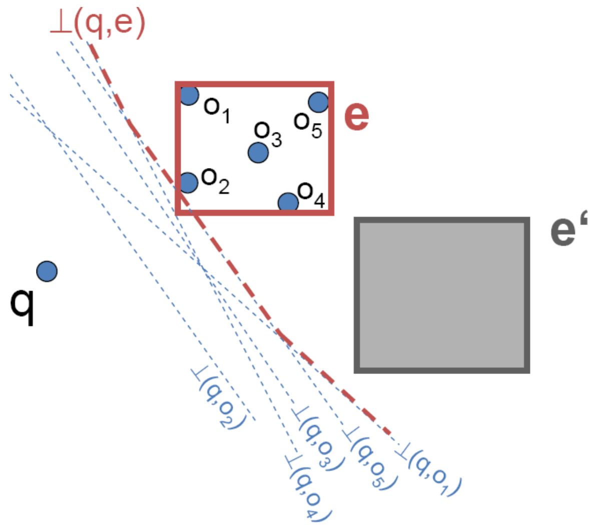


- Frage: Effiziente Repräsentation von Voronoi-Hyperebenen zwischen einem Punkt und einer Seitenregion ?
- Idee: Konservative Approximation der Hyperebenen
  Eigenschaft: $\forall p \in \perp(q,e)$: dist(q,p) = MaxDist(e,p)

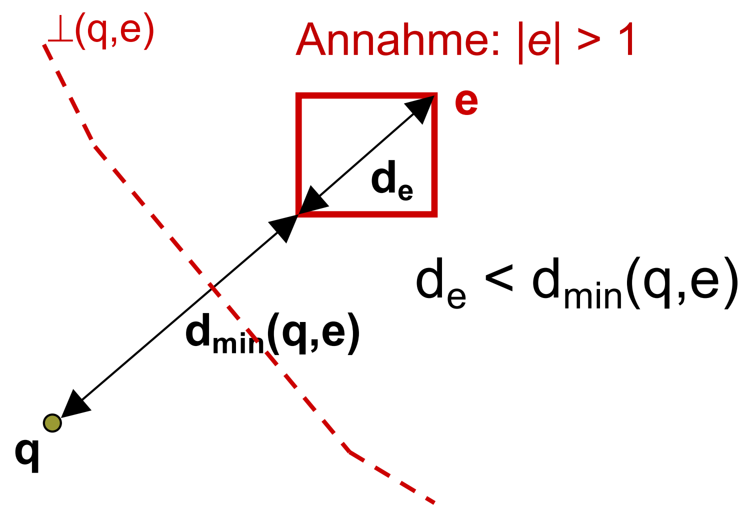
2.5 Reverse nächste Nachbarn Anfragen

- Erweiterung der Geometrischen RNN-Suche (gilt nur für Vektordaten!!!)
  [Kriegel, Kröger, Renz, Züfle: SSDBM, 2009]
  - Ziel: Pruning auf höheren Indexebenen
  - Definition von Hyper-Ebenen zwischen Anfragepunkt und Seitenregion e
  - Bilde konservative Approximation $\mathcal{H}$ aller Hyperebenen bzgl. aller Punkte innerhalb der Seitenregion e
  - Anzahl der konservativ approximierten Hyperebenen kann zum Ausschluß von Seitenregionen (Self/Mutual Pruning) verwendet werden (insbesondere auch für RkNN-Suche mit k>1)

Seite e schließt Seite e‘ aus (k=1):



Seite e schließt sich selbst aus (k=1):



$\forall o \in e'$: $o \notin$ RNN(q), da $\exists o_i \in e$: e’ hinter Hyperebene $\perp(q,o_i)$

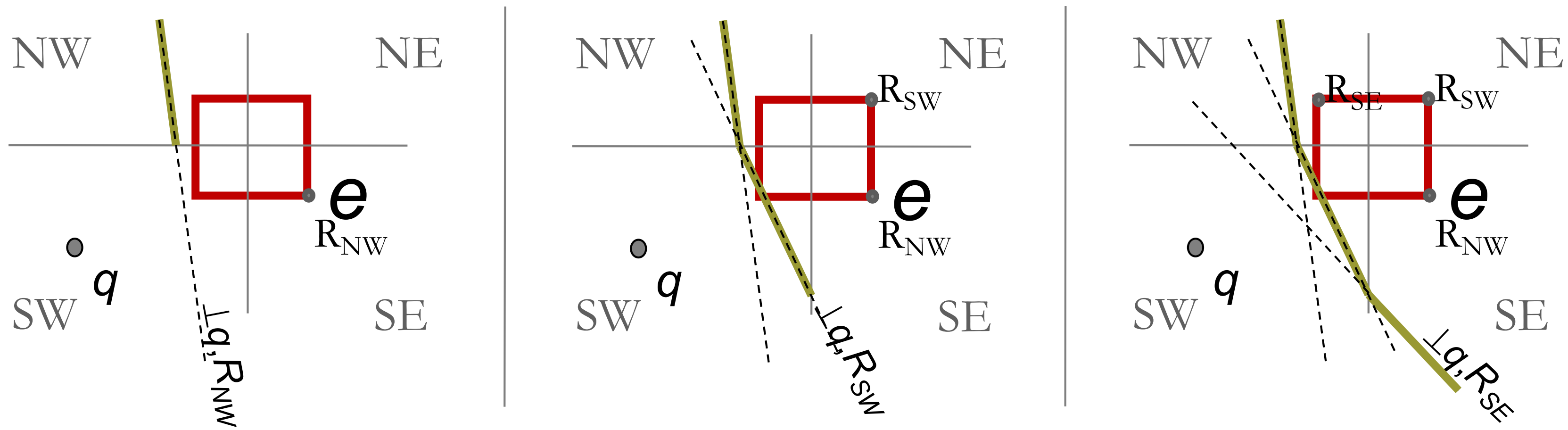
2.5 Reverse nächste Nachbarn Anfragen

- Berechnung der konservativen Approximation aller Hyperebenen zwischen Anfragepunkt q und Seitenregion e
  - Aufsplittung des Datenraums in 2^d Partitionen orientiert am Mittelpunkt der Seitenregion E (z.B. 4 Partitionen NW, NE, SE und SW in 2D Raum)
  - Für jede Partition P: Wähle Referenzpunkt R∈e, sodaß gilt:

$$\forall p \in P, \forall e \in E: d(p,e) \leq d(p,R)$$

Bemerkung: Referenzpunkt eindeutig durch die gewählte Partitionierung

  - Für jede Partition P bilde Hyperebene zwischen q und Referenzpunkt zu P



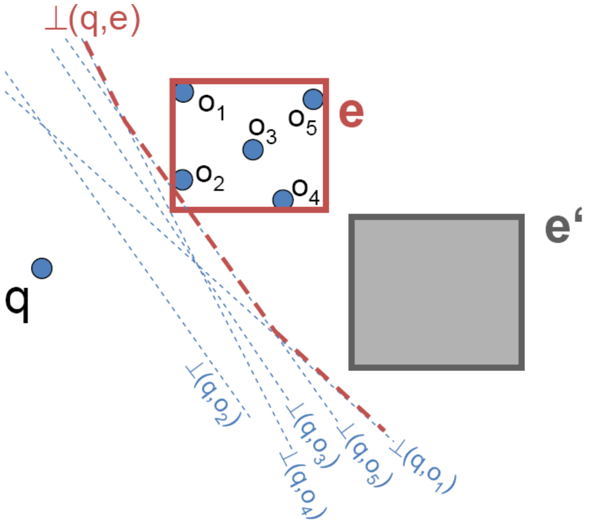
  - Die konservative Hyperebenen-Approximation wird aus 2^d-1 Hyperebenen gebildet

2.5 Reverse nächste Nachbarn Anfragen

- Erweiterung der Geometrischen R*k*NN-Suche für k > 1
  - Verwende aR-Baum anstatt R-Baum, d.h. R-Baum mit zusätzlicher Angabe der Anzahl der Punkt-Objekte |e| die innerhalb einer Seitenregion e organisiert werden.
  - Menge der Punkt-Objekte auf die sich eine konservative Approximation $\mathcal{H}$ bezieht ist zur Anfragezeit bekannt.

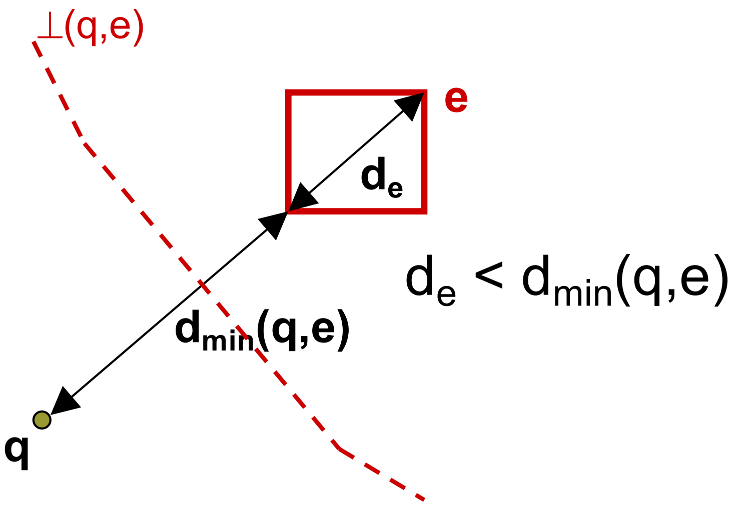
    => Anzahl N der Objekte, die näher an einem Objekt o bzw. einer Seitenregion e‘ sind als der Anfragepunkt q lässt sich einfach ermitteln
  - Prune Seitenregion e‘ (bzw. e‘ bei self pruning) falls N > k (bzw. N-1 > k)

Seite e schließt Seite e‘ aus (k>1):



$\forall o \in e'$: $o \notin$ R*k*NN(q), where $k \leq |e|$

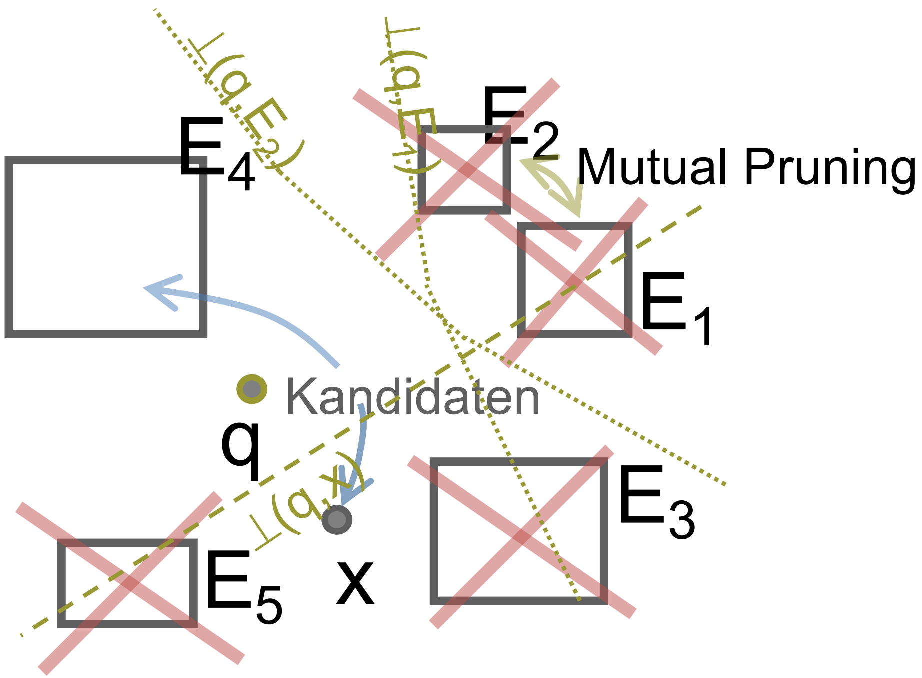
Seite e schließt sich selbst aus (k>1):



$\forall o \in e$: $o \notin$ R*k*NN(q), where $k \leq |e\text{-}1|$

2.5 Reverse nächste Nachbarn Anfragen

- Beispiel:



- Bemerkung zum „Optimalen“ Pruning:
  - Die Suche mittels Mutual-Pruning ohne Vorberechnung ist weniger selektiv als mit vorberechneten NN-Distanzen

    ==> schlechteres Pruning-Verhalten als bei (reinen) Self-Pruning-Methoden

2.5 Reverse nächste Nachbarn Anfragen

- Eigenschaften der Geometrischen R*k*NN-Suche
  - Vorteile:
    - Vollständige Flexibilität bzgl. *k*
    - Keine Zusätzliche Kosten für Änderungen im Index (Update-Kosten)
    - Pruning-Filter selektiver als bei Min/Max-Dist-Ansatz
  - Nachteile:
    - 2^d Hyperebenen pro Seitenregion müssen materialisiert werden => Overhead der Ebenenverwaltung (schlechte Performanz in höher-dimensionalen Räumen)
    - Test ob Seitenregion geprunt werden kann ist teuer (2^d Ebenen-Pruning-Tests)
- Fazit:
  - Min/Max-Dist-Ansatz hat geringeres Pruningpotential als Geometrisches Pruning
  - Geometrisches Pruning eignet sich nur für niedrig-dimensionale Räume
- Gewünscht:
  - Verfahren mit „optimalen“ Pruningpotential, das auch für hoch-dimensionale Räume effizient funktioniert

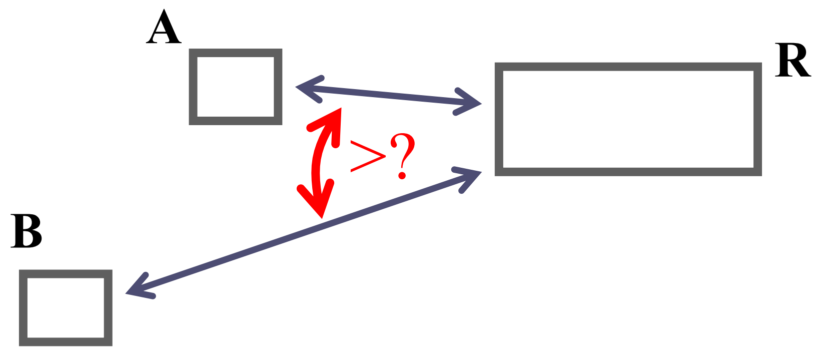
  → Welche Beziehung gilt zwischen dem Min/Max-Dist-Basierten Ansatz und dem Geometrischen Pruning Ansatz?

2.5 Reverse nächste Nachbarn Anfragen

- Generelle Problemstellung:

  Gegeben:

  Drei Punktobjekt-Approximationen A, B und R gegeben als achsenparellele Rechtecke



  Frage:

  Wie kann man effizient bestimmen ob die Objekte in R näher an den Objekten in A oder näher an den Objekten in B liegen

  → Nachbarschafts-Entscheidungskriterium [SIGMOD 10]