

**Spatial, Temporal, and Multimedia Databases**  
SoSe 2010

**Übungsblatt 3: Algorithmen zur Ähnlichkeitssuche, Mehrstufige Anfragebearbeitung**

Besprechung: 17.05.2010

**Aufgabe 3-1**     *Mehrstufige Anfragebearbeitung I*

In der Grafik (siehe Abbildung auf nächster Seite) sind sowohl die im Verfeinerungsschritt ermittelten Objektdistanzen als auch die im Filterschritt ermittelten Featuredistanzen für eine  $k$ -NN-Anfrage mit  $k=10$  eingezeichnet. Man ermittle grafisch, wie viele Objekte einem Verfeinerungsschritt unterworfen werden bei folgenden Implementierungsvarianten:

- (a) Auswertung mit Bereichsanfrage (Skript Folie 75)
- (b) Auswertung nach Priorität (Skript Folie 78).

**Aufgabe 3-2**     *Optimalität der Mehrstufigen Anfragebearbeitung bei Auswertung nach Priorität*

Die *R-Optimalität* einer mehrstufigen  $k$ -NN Anfragebearbeitung sei folgendermaßen definiert:

Definition (R-Optimalität): Ein mehrstufiger  $k$ -NN-Anfragealgorithmus ist *R-optimal*, wenn im Filterschritt nicht mehr Antwort-Kandidaten erzeugt werden als notwendig.

Wie kann man zeigen, daß die mehrstufigen Anfragebearbeitung bei Auswertung nach Priorität (Algorithmus siehe Skript S. 78) keine *false drops* erzeugt (d.h. keine Ergebnisse verloren gehen) und *R-optimal* ist. (Bem.: Eine Skizze der Beweisidee genügt).

**Aufgabe 3-3**     *Basisalgorithmen zu Ranking Queries*

Überlegen Sie sich drei Basis-Algorithmen für die Ranking-Query ohne Indexunterstützung. Berücksichtigen Sie hierbei, daß nicht die *gesamte* Datenbank in den Arbeitsspeicher paßt. Die Anzahl der `get-next` Aufrufe umfaßt jedoch potentiell die gesamte Datenbankgröße.

Diskutieren Sie die alternativen Lösungen unter Berücksichtigung folgender Benutzerszenarien:

- (a) Ein Benutzer ruft typischerweise zwischen 10 und 100 Anfrageergebnisse ab.
- (b) Ein Programm, das die Anfrageergebnisse weiterverarbeitet, ruft zwischen 10% und 100% aller Datenbankobjekte ab.

object and  
filter distances

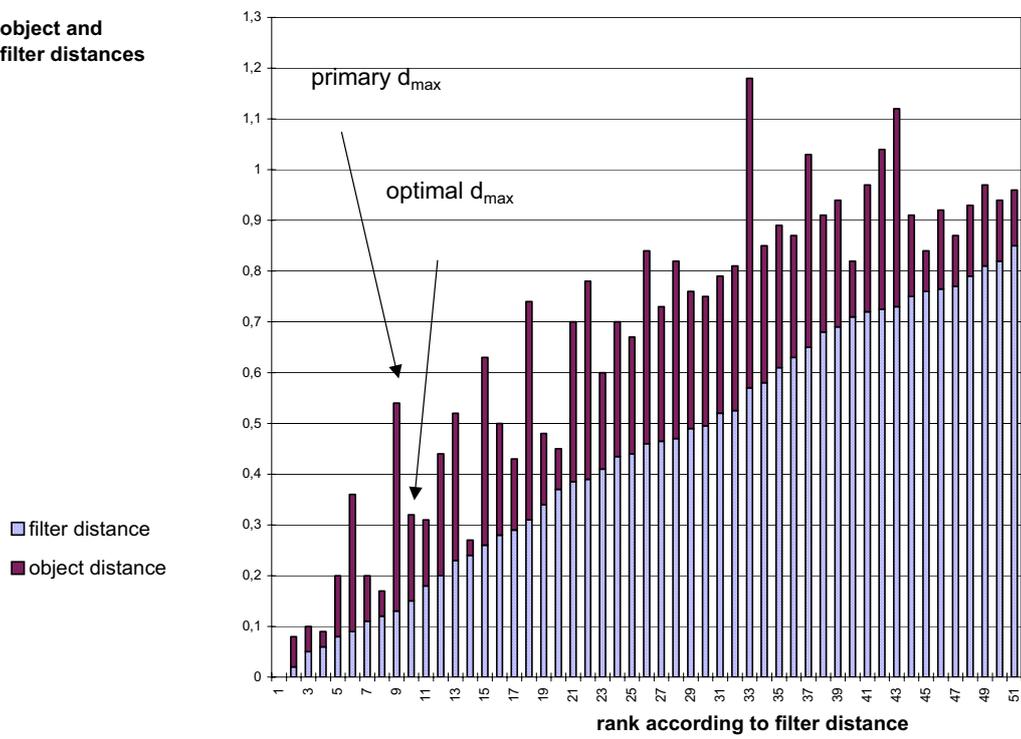


Abbildung 1: Bild zu Aufgabe 3-2