Alexander Hinneburg,
hinneburg@informatik.uni-halle.de

3.12.2005

Übung zur Vorlesung „Data Mining in Datenbanken“

Übungsblatt 7[1]

Häufige Item-Mengen und Clusteranalyse
Abgabe am 8. bzw. 15.2005

Aufgabe 6.6 Implementieren Sie den A-priory Algorithmus mit möglichst kleiner Laufzeit und Haupspeicherverbrauch. Vermeiden Sie große Datenstrukturen doppelt im Hauptspeicher zu halten. Testen Sie Ihre Implementierung an den nominalen Wetterdaten mit $minsupport = 2$, die mit dem WEKA Paket mitgeliefert werden. Bestimmen Sie die Anzahl der häufigen Item-Mengen der Größe 1, 2, 3, 4, 5 für $minsupport = 2$ und 3.

Aufgabe 7.1 Aus dem EVV wurden die Interessen aller Studenten in anonymisierter Form extrahiert. Die Daten liegen als zwei Tabellen vor. Tabelle 1 enhält die Spalten `student_id` und `vorlesungs_id` und Tabelle 2 ordnet der `vorlesungs_id` den Name der Vorlesung zu (`vorlesungs_name`). Die Daten finden Sie auf der Vorlesungsseite in den entsprechenden Text-Dateien. Die Spalten sind durch "getrennt.

Generieren Sie aus Tabelle 1 eine Liste von „Transaktionen“, wobei eine Transaktion die Liste der `vorlesungs_id`'s enthält, die ein Student in diesem Semester besucht. Beschreiben Sie Ihre Arbeitsschritte. Finden Sie mit Hilfe des Apriory Algorithmus heraus für welche Kombinationen von Vorlesungen sich Studenten im WS 05/06 häufig interessieren. Geben Sie die häufigsten drei Kombinationen der Größe drei, vier und fünf mit den absoluten Häufigkeiten aus.

Aufgabe 7.2 Ziel dieser Aufgabe ist es, eine Implementierung des k-Means Algorithmus in SQL vorzubereiten. Sie sollen SQL-Anfragen zur Berechnung der initialen Repräsentanten mittels zufälliger Zuordnung der Instanzen zu den Repräsentanten sowie zu Berechnen der neuen Repräsentanten (k-Means Schritt 2) erarbeiten. Sie können hier auf Ihre Erfahrungen aus der SQL-Implementierung für Histogramme zurückgreifen.

In dieser Aufgabe sollen die Iris-Daten ohne die Klassenattribute verwendet werden, d.h. die Tabelle hat die Spalten `pid` (ID für einen Datenvektor), `sepal_length`, `sepal_width`, `petal_length` und `petal_width`. Zusätzlich sollen Sie noch eine Tabelle `w` mit den gleichen Daten-Attributen wie in den Iris-Daten anlegen, in der die k Repräsentanten gespeichert werden. Um die Repäsentanten aller Iterationen in dieser Tabelle zu speichern, soll `w` um die Integer-Attribute `wid` (von $1, \ldots, k$) und `iter` erweitert werden, welche zusammen den Primärschlüssel bilden.

Der k-Means Algorithmus erfordert in Schritt zwei die Berechnung des Mittelpunktes der zu einem Repräsentanten zugeordneten Datenvektoren. Diese Operation kann mittels mehrerer Durchschnittsbildungen (eine für jedes Attribut) in SQL realisiert werden.

[1] Achten Sie auch auf die Form Ihrer Lösungen, z.B. daß jedes Bild eine Bildunterschrift hat. Bitte geben Sie keine losen Blätter ab. Um die Übungen in angemessener Zeit zu diskutieren, bereiten Sie neben Ihrer Lösung ein kleine Präsentation vor, mit der Sie dann Ihre Ergebnisse zeigen können.

Alle Positionen der neuen Repräsentanten sollen in einem Schritt durch eine `group by` Anweisung berechnet werden. Gruppiert werden die Datenpunkte nach der ID (`wid`) des zugeordneten, nächsten Repräsentanten.

Die Datenpunkte werden zu dem Repräsentanten mit der minimalen euklidischen Distanz zugeordnet. Die minimale Distanz zum Repräsentanten kann auch mittels `group by` berechnet werden. Dann aber müssen Sie müssen Sie mit der ID des Repräsentanten mit der minimalen Distanz weiterrechnen. Um diese ID zu bekommen müssen Sie einen sogenannten Back-Join machen, der im folgenden Beispiel abstrakt erläutert wird.

Beim Back-Join soll aus einer Tabelle mit den Spalten `id` und `value` die Zeilen-ID mit dem minimalen `value` gesucht werden. Beachten Sie, daß es mehrere Zeilen mit dem minimalen `value` geben kann. Um dieses Unentschieden aufzulösen, wird aus den Zeilen mit dem Minimum `value` diese mit der kleinsten `id` herausgesucht.

```
data(id,value)
select min(id)
from (
  select min(value) as min_value
  from data
  ) a, data
where a.min_value=data.value
```

7.3 (zum 15.12.05) Implementieren Sie den k-Means Algorithmus mittels wiederholter SQL Anfragen. Hier sollen Sie auf die Ergebnisse aus 7.2 zur Initialisierung und zum Berechnen der neuen Repräsentanten zurückgreifen und diese mit `Insert` Statements verknüpfen. Neu müssen Sie eine Anfrage zu Berechnung des Quantiserungsfehlers D entwickeln. Koordinieren Sie die Anfragen mittels eines Skriptes, welches den Oracle-Instant-Client mit entsprechenden Anfragen aufruft. Überlegen Sie zum 8.12. mögliche Schwierigkeiten, die dann diskutiert werden können.