**LighthouseUCC**

LighthouseUCC ist ein von uns entwickelter Algorithmus und baut nicht direkt auf anderen

Algorithmen auf, enthält aber Einflüsse von DUCC.

**1 Genereller Ablauf**

Wir erzeugen PLIs für jede Spalte. Wir berechnen für jede Spalte die Anzahl distinkt

Werte als Score. Dann löschen wir alle Spalten, die bereits Unique sind. Die verbliebenen

Kandidaten kommen in eine Liste.

Mainloop (Solange Kandidaten in unserer Liste vorhanden sind):

Wir sortieren die Kandidaten anhand ihres Scores. (PriorityQueue) Wir wählen nun den

Kandidaten mit dem höchsten Score und testen ihn auf Uniqueness.

Wenn er Unique ist:

* Füge ihn in die Unique Liste ein
* Entferne Supersets dieses Kandidaten aus der Unique List
* Außerdem erzeugen wir direkte Subsets des Kandidaten und fügen diese in die Kandidaten

Liste ein.

Sonst:

* Vom Kandidaten werden jetzt mit einzelnen Spalten Spaltenkombinationen gebildet.
* Dabei werden nur neue Spaltenkombinationen (keine Permutationen) berücksichtigt.
* Auch erzeugen wir keine Spaltenkombination für die wir bereits kleinere UCCs gefunden haben.
* Diese neuen Spaltenkombinationen sind neue Kandidaten in unserer Liste.

**2 Komplexität**

* Der Algorithmus bezieht Informationen über die Distinktheit der einzelnen Spalten ein
* Viele Kandidaten fallen durch das Wandern nach oben im Baum weg

**3 NULL-Semantik**

*null != null*

* potentiell hohe Wahrscheinlichkeit Teil einer minimalen UCC zu sein
* durch hohen Score früh in unserem Algorithmus behandelt

*null == null*

* potentiell niedrige Wahrscheinlichkeit Teil einer minimalen UCC zu sein
* durch niedrigen Score spät in unserem Algorithmus behandelt

**4 Performance**

Unsere Messungen haben wir auf einem Windows PC mit i7 4500U 1,80GHz durchgeführt. Der Anwendung standen 2GB RAM zur Verfügung.

|  |  |  |
| --- | --- | --- |
| Datensatz | Anzahl UCCs | Zeit |
| ncvoter-1k.csv | 98 | 1s 206ms |
| adult.csv | 0 | 306ms |
| WDC\_astronomical.csv | 4 | 25ms |
| WDC\_planets.csv | 7 | 29ms |
| WDC\_satellites.csv | 4 | 44ms |
| ncvoter.csv (~1Gb) | Out of heap |  |

**5 Probleme**

* Falls die Uniques sehr groß sind, dann werden aktuell noch zu viele Kandidaten erzeugt.
* Beim Erzeugen der direkten Subsets im Falle eines gefundenen Uniques sucht in die Breite anstatt in die Tiefe, wodurch wiederum zu viele Kandidaten erzeugt werden
* Dies führt zu längeren Laufzeiten oder gegebenenfalls zu einem Out-of-heap Fehler