Verslag BDA opdracht 1

Sil Vaes en Maarten Evenepoel

October 12, 2021

In deze opdracht was de bedoeling dat we een maximal frequent itemset algoritme ging implementeren. Hier hebben we het a-priori algoritme met hashing geïmplementeerd, dus het PCY algoritme.

Het inlezen van de dataset in het XML-formaat wordt gedaan met behulp van regexes, dus de XML library in de standard library van Python is niet gebruikt. We hebben bewust deze keuze gemaakt omdat dit ons eenvoudiger leek, want we moesten enkel de entries tussen bepaalde tags matchen en dan enkel the authors. Dus we zijn tot de conclusie gekomen dat een hele XML parser wat overkill was.

De dataset wordt ook niet in een keer ingelezen, maar in chunks van x megabytes, waar x een commandline argument is.
Het programma wordt als volgt opgeroepen:

```
python main.py {{dataset}}.xml {{chunk_size}}
```

Waar {{dataset}} en {{chunk_size}} het bestand dat de dataset bevat en de chunk size waarmee het bestand wordt ongelezen zijn respectievelijk. Ook kan men een optionele `--make\_testfile` flag toevoegen om een kleinere testfile van de grotere {{dataset}} file te maken.

Bij k's van één tot en met drie wordt er een ander algoritme gebruikt om kandidaten te vinden bij k's van hoger dan drie. Hier is voor gekozen omdat in praktijk het snelle algoritme te geheugenineffiçiënt bleek te zijn en het meer geheugenefficiënt algoritme te traag. Dus hebben we besloten een compris te maken aan de hand van emprische ervaring.

Het programma dat geschreven is in deze assignment werkt op kleinere subsets van de dataset, we hebben het geprobeerd te runnen op de gehele dataset, maar het geheugen van onze laptop geraakten te snel vol. Onze laptops hadden maar

een geheugen van 8 GB. We hebben het geprobeerd te optimalisen met behulp van hashing. Dit verbeterde de situatie, maar het geheugen geraakte alsnog vol.