

Metódy zlepšenia lokálnosti webového grafu

Report za letný semester

Mojím cieľom v letnom semestri bolo prakticky implementovať a pretaviť teoretické základy nadobudnuté v predchádzajúcom období do pipeline na kompresiu a analýzu reálnych webových grafov a následne ho otestovať na dumpoch slovenskej Wikipédie. Pôvodná ambícia samostatne implementovať všetky algoritmy numerickej lineárnej algebry „od základov“ sa ukázala ako extrémne neúnosná, preto boli napokon na realizáciu výpočtov použité optimalizované implementácie z knižníc jazyka Python.

V rámci projektu boli implementované nasledujúce programy a skripty:

- `build_graph.py`
Skript zostrojí maticu susednosti z informácií získaných zo surových dumpov Wikipédie a zapíše ju na disk v podobe riedkej matice.
- `compress_graph.py`
Hlavný výpočtový modul celého reťazca, ktorý realizuje samotné spektrálne usporiadanie uzlov s cieľom zlepšiť lokálnosť grafu. Vykonáva randomizovanú SVD dekompozíciu a následnú vlastnú dekompozíciu matice VV^T pomocou Lanczosovho algoritmu. V tomto kroku sa naplno využíva vlastnosť, že matica VV^T nemusí byť nikdy explicitne vyjadrená a uložená v pamäti; stačí postupne počítať iba vektory tvoriace bázu Krylovovho podpriestoru pre maticu VV^T , čo výrazne znižuje celkovú pamäťovú náročnosť. Po úspešnej spektrálnej klasterizácii skript zapíše graf v skomprimovanom tvare (s využitím delta a referenčného kódovania) do súboru.
- `rank_pagerank.py`
Skript implementuje algoritmus PageRank. Pre výpočet jednej matematickej iterácie program nikdy nepotrebuje držať celú maticu v pamäti RAM, čo je umožnené tým, že štruktúru grafu je možné dekodovať postupne po jednotlivých vektoroch.

Pre pohodlnejšie spúšťanie, testovanie a správu týchto modulov boli vytvorené aj pomocné skripty:

- Skript `launch.py` slúži ako orchestrátor a spúšťa celý výpočtový reťazec sekvenčne za sebou. Skripty `launch.py`, `build_graph.py`, `compress_graph.py` a `rank_pagerank.py` akceptujú jeden argument z príkazového riadka – cestu ku

konfiguračnému súboru parametrov. Príklady konfigurácií sú uložené v adresári configs.

- Skript `download_dumps.py` automaticky stiahne najnovšie dumpy z repozitárov Wikimedia, ktoré následne slúžia ako vstup pre pipeline. Skript `compress_graph.py` je samozrejme po drobných modifikáciách možné použiť aj na iné typy sieťových grafov.

Dosiahnuté výsledky a zhodnotenie

Skript `compress_graph.py` úspešne vyprodukoval súbory popisujúce skomprimovaný graf (metadáta grafu a samotnú maticu susednosti), ktorých sumárna veľkosť dosiahla 23,7 MiB. Pre porovnanie, surový súbor produkovaný modulom `build_graph.py` mal veľkosť až 121 MiB. Tieto kompresné výsledky sú priložené a dostupné v GitHub repozitári. Počas testovania bolo taktiež možné jasne pozorovať merateľné zväčšenie lokálnosti grafu (a teda aj priamu efektívnosť kompresie) spolu s rastom parametrov v smeroch, ktoré teoreticky ovplyvňujú úspešnosť kódovania. Skript sa teda správa plne podľa očakávaní. Napríklad spustenie reťazca s konfiguráciou `basic` viedlo k 86 000 úspešným referenciám v okne, avšak spustenie s konfiguráciou `extreme` vygenerovalo až 108 000 úspešných referencií.

Na základe týchto dát predpokladám, že pri nasadení na štrukturálne väčšie grafové siete bude mať navrhnutý algoritmus ešte vyššiu účinnosť.

Výsledok algoritmu PageRank sa zhoduje s očakávaním a, bohužiaľ, žiadne prekvapivé udalosti nepriniesol.