KIV/VSS

1.9. – Generování náhodných čísel

Gaussovské rozdělení

Miroslav Liška – A17N0081P

topiker@students.zcu.cz

9.12.1992

28. listopadu 2017

1 Zadání

1.1 Teoretické pozadí

Zejména při fyzické aktivitě nebo příjmu potravy dochází k výrazné změně koncentrace glukózy. V zadaných datech máte několik měření koncentrací glukózy v intersticiální tekutině [mmol/l]. Jsou vzorkovaná po 5 minutách tzv. systémem CGMS. U každého měření je časová značka a měření jsou rozdělena do segmentů, které trvají od několika hodin do několika dní. Vaším úkolem je identifikovat cca 3 - 5 významných kolísání koncentrací glukózy během dne (uvažuje se 3 jídlo a 2 fyzická zátěž). Při identifikaci si můžete pomoci i časovými značkami, případně naměřenou koncentrací v krvi, která by po jídle a při fyzické aktivitě měla být taktéž zvýšená.

V běžném životě si pacient údaje o jídle a fyzické aktivitě zadává manuálně a ještě s chybou. Systém automatické detekce by tuto chybu redukoval a napomohl tak lepší léčbě pacienta. Práce tedy není "šuplíková", ale má praktické využití (cca každý 11. člověk má diabetes a polovina z nich o tom ani neví, protože diabetes nebolí, dokud není pozdě).

Změny koncentrace glukózy lze nejjednoduššeji detekovat jako ohodnocené plovoucí okno - tj. fixní časový úsek, který bude "plout"v čase segmentu od jeho počátku až na konec. Ohodnocení okna může být součet rozdílů koncentrací glukózy v daném okně. S algoritmem lze dále experimentovat, např. velikost okna a mezní ohodnocení (tj. od kdy je okno považováno za významnou změnu koncentrace glukózy) lze určovat např. pomocí Diferenciální evoluce, nebo jiným algoritmem - může to být i 2D půlení intervalu. Vlastní invenci při vývoji detekčních algoritmu se meze nekladou.

1.2 Verze úlohy

Zpracujte úlohu alespoň ve dvou verzích ze tří možných:

- Paralelní program pro systém se sdílenou pamětí
- x86 CPU + OpenCL/C++ AMP GPGPU
- Paralelní program pro systém s distribuovanou pamětí

1.3 Data

Naměřené hodnoty jsou uloženy ve formátu SQLite verze 3. Konkrétně jsou uloženy v tabulce measuredvalue. Požadované hodnoty najdete ve sloupci ist, který vyjadřuje koncetraci v intersticiální tekutině v [mmol/l]. Čas měření je zanesen ve sloupci measuredat, a je ve formátu ISO 8601. Data jsou seskupena do tzv. segmentů, viz sloupec segmentid. Naměřená data zpracovávejte vždy po celých segmentech. Jméno segmentu lze dohledat v tabulce timesegment a jméno pacienta analogicky v tabulce subject.

1.4 Výstup

Na stdout vypište získané statistické ukazatele jako tabulku v csv formátu. Zároveň vygenerujte grafický výstup ve formátu SVG (pro každý segment jedno SVG), ve kterém graficky znázorníte změny koncentrace glukózy považované za příjem potravy, fyzickou aktivitu, apod. Implentujte přepínač, který buď segment vykreslí v celé jeho délce, anebo ho bude zalamovat po 24 hodinách - tj. osa X (čas) bude mít hodnoty od 00:00 do 23:59. V takovém případě by mohlo být vidět, např. zda pacient snídán či večeří pravidelně - což je také možná nápověda pro detekční algoritmus.

1.5 Další statistiky

Program také spusťte s jedním vláknem/procesem a změřte čas výpočtu sériovým kódem a čas výpočtu paralelizovaným kódem (pro všechny verze paralelizovaného kódu zvlášť). Z těchto hodnot vypočítejte následující ukazatele:

- Amdahlův zákon, f – čas sériově prováděné části kódu
- Gustafsonův zákon, a – část kódu, kterou nelze paralelizovat
- Karp-Flattova metrika, e – část sériově prováděného kódu

2 Analýza

2.1 Detekce změn koncentrace glukózy

O chování koncentrace glukózy v krvi víme (informace z přednášek), že pokud pozorovaný subjekt zkonzumuje nějakou potravinu, konentrace vzroste. Pokud je subjektem vynaložena nějaká aktivita, koncentrace typicky mírně vzroste a pak začne klesat. Tyto akty pak v datech generují významné kolísání, jejichž detekce je cílem práce. Dále je z přednášky známo, že kolísání trvá typicky tři hodiny s tím, že nárůst trvá hodinu a následné klesání pak dvě hodiny.

Jedním z možných řešení detekce je nalezení lokálních extrémů. Následně se pro každý extrém vezmou spojitě data začínající před extrémem a po extrému o nějaké velikosti. Pro výběr nejlepších výsledků je nutné získané intervaly ohodnotit. Dále je potřeba nějakým způsobem naložit s překrývajícími se intervaly, například jejich sloučením či vyřazením horšího.

Dalším řešením může být evoluční genetický algoritmus. Data jsou rozdělena náhodně na intervaly o pevné velikosti a pro každý interval je vypočítána jeho fitness funkce. Následně se vybrané intervaly posunou a spočítá se jejich fitness funkce a tím vznikne nová generace intervalů. Podle chování genetického algoritmu se s novou generací patřičně naloží. Posouvání probíhá do té doby, dokud nevznikne nejlepší generace intervalů, které jsou pak detekovaným kolísáním.

2.1.1 Zvolené řešení

Pro detekci kolísání jsem se rozhodl použít algoritmus posuvného okénka. Princip spočívá v tom, že se napříč daty iteruje tzv. okénkem o velikosti n. Každá iterace posune okénko o jedno měření dál. Každé okénko je ohodnoceno funkcí. V implementaci je jako funkce zvoleno rozdíl sousedních hodnot na druhou. Na druhou z toho důvodu, aby nebylo okénko ohodnoceno záporně.

Jakmile je spočítáno ohodnocení všech okének, je potřeba vybrat pouze ty nejlepší a nějakým způsobem naložit s okénkami, které se překrývají. Při výběru významějších okének jsem se rozhodl vybrat pouze ty, jejichž ohodnocení je lepší, než průměrná hodnota. Po výběru významějších okének je možné, že se budou jednotlivé intervaly překrývat. Je předpokladem, že ve vybraná okénka budou reprezentovat části s významným klesáním či nárůstem. Pokud se tedy intervaly překrývají, je vhodné je spojit. Pro takto spojená okénka znovu spočítáme jejich ohodnocení a následně se vezme n nejlepších.

2.2 Načtení uložených dat

Naměřené hodnoty jsou ve formátu SQLite verze 3. Pro přístup k datům je tedy vhodné přistoupit za pomocí SQL dotazů. Při analýze bylo zjištěno, že ne všechny segmenty uvedené v tabulce segmentů mají i naměřená data a naopak nějaká naměřená data mají id segmentu, které není v tabulce segmentů. Při načtení dat z databáze budou tedy zvolena pouze ta data, jejichž id segmentu odpovídá tabulce segmentů. Pro práci hledání kolísání bude nutné, aby data

byla seřazena podle data měření (measuredate). Datum měření je uloženo ve formátu ISO 8601. Podle doporučení je dobré převést datum na formát, kdy hodnota 1.0 představuje datum 1.1.1900.

2.3 Paralelizace

Na obrázku 1 je možné vidět průběh výpočtu a interakcí programu. Slovně byl popsán v sekci 2.1.1

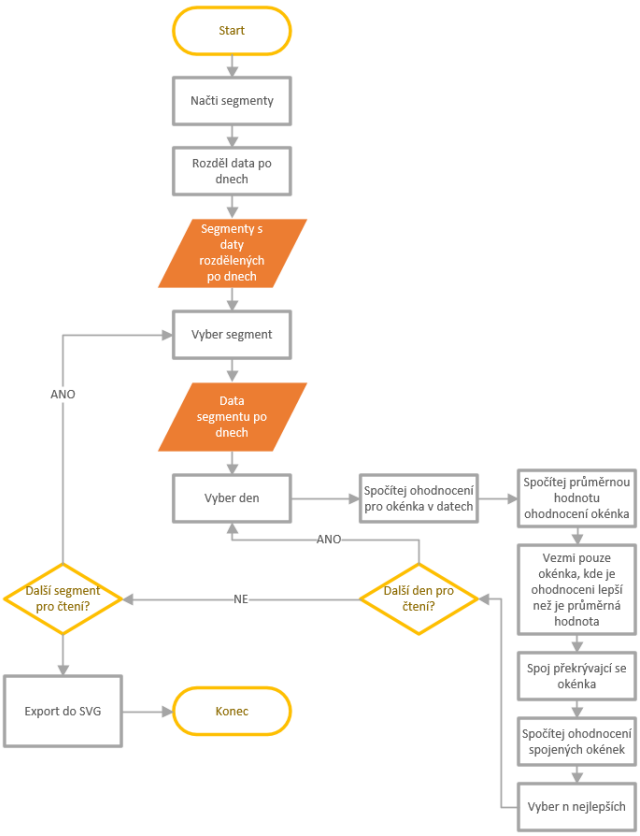
Pro parelelizaci této úlohy jsem si vybral paralelizaci se sdílenou pamětí a paralelizaci na GPGPU.

Pravděpodobně nejlepším řešením bude paralelizace na úrovni segmentů, tedy detekce kolísání jednotlivých segmentů bude probíhat současně. Důvodem výběru tohoto místa je, že se jedná o práci s větším množstvím nezávislých dat, tedy by režie spojená s paralelizací nemusel být zpomalující. Další možností paralelizace je paralelizace uvnitř segmentu na úrovni jednotlivých dní. Jednalo by se tak o paralelizaci uvnitř paralelizace (segment a jednotlivé dny). Vzhledem k tomu, že počet dat uvnitř jednotlivých dní je nízký (maximálně několik stovek), je předpokladem, že tato paralelizace přínos nepřinese. Tento přístup ale bude v rámci experimentu implementován.

Vzhledem k charakteru zvoleného algoritmu (mnoho podmínek, porovnávání) a obecně nízkému počtu dat, se kterými se provádí matematické operace, je předpokladem, že paralelizace na GPGPU bude spíše zdržující.

3 Řešení

4 Závěr



Obrázek 1: Flowdiagram programu