Języki i narzędzia programowania 2 - CUDA

Wariant I - GWAS (Genome-Wide Associacion Study)

Założenia

Dany jest system informacyjny składający się N obiektów opisywanych K zmiennymi dyskretnymi przyjmującymi wartości ze zbioru {0,1,2} oraz zmienną decyzyjną D. Wartość 0 oznacza brak występowania mutacji na pewnej pozycji w genomie, wartość 1 oznacza występowanie mutacji w jednym allelu, wartość 2 na obu allelach. Zmienna decyzyjna przyjmuje dwie wartości np. 0/1. i można ją powiązać z efektem fenotypowym - np. występowanie pewnej choroby.
Liczba zmiennych K może wynosić do 20 000 000, liczba obiektów N jest dowolna. Na potrzeby zadania można przyjąć, że liczba obiektów w klasie decyzyjnej nie przekracza 30 000.
W danych dostarczonych do testów K = 100 000, N = 5 000, ale algorytm nie może polegać na tych wartościach.

Tabela 1. Przestrzeń możliwości dla dwuwymiarowego klasyfikatora GWAS

i / j	V_j	V_j	V_j
V_i			
V_i			
V_i			

Celem jest zidentyfikowanie zmiennych, które są powiązane ze zmienną decyzyjną. Zmienna opisowa jest powiązana ze zmienną decyzyjną jeżeli znajomość zmiennej opisowej może zmniejszyć entropię informacyjną układu w sposób statystycznie istotny.
Entropia informacyjna układu :

$$H = -\sum_{A} p^A \log p^A = -p \log p - (1-p)\log(1-p)$$

A przyjmuje wszystkie wartości zmiennej decyzyjnej, czyli dla dwóch klas możemy wyrazić entropię przez jedno prawdopodobieństwo - np. prawdopodobieństwo klasy chory w badanym zbiorze. Znak minus wynika z konwencji - entropia jest dodatnia, a logarytm prawdopodobieństwa jest liczbą niedodatnią.

Dla układu, w którym znamy wartość zmiennej opisowej, przyjmującej wartości k={0,1,2} entropia informacyjna wyraża się jako:

$$H = -\sum_{A}\sum_{k} N_k p_k^A \log p_k^A = -\sum_{k} N_k \left\{ p_k \log p_k + (1-p_k)\log(1-p_k) \right\}$$

Gdy znamy wartości dwóch zmiennych opisowych przyjmujących odpowiednio wartości k={0,1,2}, l={0,1,2} entropia wynosi:

$$H = -\sum_{A}\sum_{k}\sum_{l}\sum_{m} N_{klm} p_{klm}^A \log p_{klm}^A = -\sum_{k}\sum_{l}\sum_{m} N_{klm} \left\{ p_{klm} \log p_{klm} + (1-p_{klm})\log(1-p_{klm}) \right\}$$

a w przypadku trójwymiarowym {k,l,m} = {0,1,2},

$$H = -\sum_{A}\sum_{k}\sum_{l} N_{kl} p_{kl}^{A} \log p_{kl}^{A} = -\sum_{k}\sum_{l} N_{kl}\left\{ p_{kl} \log p_{kl} + \left(1 - p_{kl}\right)\log\left(1 - p_{kl}\right)\right\}$$

W powyższych wzorach $\left\{N_k, N_{kl}, N_{klm}\right\}$ to liczba obiektów obu klas w odpowiednim przedziale.

Przyrost informacji związany z posiadaniem wiedzy o wartościach zmiennych opisowych jest dany dla jednej, dwóch i trzech zmiennych jako:

$$IG(p \mid x) = H(p) - H(p \mid x)$$
$$IG(p \mid x,y) = H(p) - H(p \mid x,y)$$
$$IG(p \mid x,y,z) = H(p) - H(p \mid x,y,z)$$

Przyrost informacji związany z posiadaniem wiedzy o wartościach dwóch zmiennych opisowych w porównaniu ze znajomością jednej zmiennej wynosi:

$$GIG_{1\to 2}(p \mid x,y) = IG(p \mid x,y) - \max\left\{IG(p \mid x), IG(p \mid y)\right\}$$

Odpowiednio dla dwóch i trzech zmiennych mamy

$$GIG_{2\to 3}(p \mid x,y,z) = IG(p \mid x,y,z) - \max\left\{IG(p \mid x,y), IG(p \mid y,z), IG(p \mid x,z)\right\}$$

Obliczenie prawdopodobieństw we wzorach na entropię.

Wzory podane powyżej odpowiadają przypadkowi idealnemu. W wypadku rzeczywistych danych może okazać się, że w wielu przypadkach prawdopodobieństwa wyliczone empirycznie mogą przyjąć wartości 1 lub 0. Takie wartości mogą bardzo mocno wpłynąć na wyniki całości. Dlatego prawdopodobieństwo we wzorze liczymy jako kombinację liniową prawdopodobieństwa a priori i prawdopodobieństwa wyliczonego z obserwacji. Jeżeli określimy przez *p* prawdopodobieństwo *a priori* klasy 0, N_0 i N_1 to odpowiednio liczba zliczeń obiektów klasy 0 i 1 to skorygowane prawdopodobieństwo klasy zero p_0 liczymy jako

$$p_0 = \frac{N_0 + p}{N_0 + N_1 + 1}$$

Zadanie:

Wyznaczyć **P** par zmiennych z największym $GIG_{1->2}$, gdzie P ~= 10 000.

Algorytm

Dla każdej możliwej pary zmiennych musimy policzyć wartość $GIG_{1->2}$ i zapisać na liście par jedynie takie dla których otrzymana wartość jest wyższa od pewnego poziomu odcięcia:

```
for (i=0; i < K;i++) {
      for j=i+1; j<K; j++) {
            if GIG_12(i,j)>Threshold Add_To_List(i,j);
      }
}
```

Żeby ustalić jaki powinien być poziom odcięcia (parametr *Threshold*) należy wykonać próbny przebieg dla niewielkiego zrandomizowanego podzbioru danych.
W tym celu stosujemy następującą procedurę:

- Wybieramy ze zbioru losowo 10% zmiennych. Losowo zamieniamy wartości zmiennych między obiektami. Liczymy dla wszystkich możliwych par zmiennych w zrandomizowanej próbie $GIG_{1->2}$. Budujemy histogram wartości uzyskanych w próbie losowej.
- Wybieramy poziom odcięcia przy którym nasza lista będzie liczyła 0.01 * P elementów jako ostateczny poziom odcięcia.

Taka procedura powinna doprowadzić do znalezienia empirycznie poziomu odcięcia, dla którego powinniśmy uzyskać około P par zmiennych na liście wyników. Dla tych par możemy potem przeprowadzić dokładniejszą analizę statystyczną.

Zadanie bonusowe:
Napisać analogiczny algorytm dla przypadku trójwymiarowego.

Wskazówki

1. Redukcja zużycia pamięci - wartości zmiennej decyzyjnej można zapisać w postaci wektora bitowego 0/1. Wartości zmiennych opisowych można zapisać w postaci wektorów bitowych - albo trzech wektorów bitowych przyjmujących wartości 0/1, każdy wektor odpowiadający wartości {0,1,2} dla danego obiektu (dla każdego obiektu jeden i tylko jeden z tych wektorów będzie miał zapaloną wartość 1); albo w postaci wektora bitowego, w którym dwa kolejne bity kodują wartość zmiennej dla obiektu.
2. Tablicę par zmiennych możemy podzielić na kwadratowe kafelki o rozmiarze np. 16x16 albo 32x32 (wielokrotność 8). Każdemu kafelkowi odpowiada blok wątków. Jeden wątek liczy klasyfikator dla jednej pary zmiennych. Wartości zmiennych potrzebnych wszystkim wątkom wczytujemy do pamięci współdzielonej.

Przykładowy uogólniony algorytm uogólniony dla karty:

Podziel przestrzeń przeszukiwań na kafelki o rozmiarze N_k x N_k
Dla każdego kafelka
    Dla każdej pary zmiennych (j,k)
        znajdź $GIG_{1->2}$
        sprawdź czy jest powyżej poziomu odciecia
        jeżeli jest
            zapisz wynik w postaci *{j,k,v1,v2,s}*
        fi
    done
done

Wariant II - All Relevant Feature Selection (ARFS)

Podobnie jak w wariancie I zmienna decyzyjna przyjmuje wartości 0/1. Natomiast zmienne opisowe są liczbami rzeczywistymi z dowolnego rozkładu. Możemy jednak zastosować podobny schemat określania, które zmienne są ważne jak w przypadku zmiennych dyskretnych w GWAS. W tym celu należy przeprowadzić dyskretyzację zmiennych - zamienić wartości rzeczywiste na kilka wartości dyskretnych. Odpowiada to podzieleniu każdej zmiennej na kilka klas wartości. Przykładowo, wprowadzenie jednego podziału odpowiada wprowadzeniu dwóch klas -mały/ duży. Wprowadzenie 3 klas mała/średnia/duża. Wprowadzenie 5 klas to podział (bardzo mała)/ mała/średnia/duża/(bardzo duża) itd.

Zasadnicza różnica polega na tym, że w wypadku GWAS zmienne są od razu w postaci klas. Tymczasem w wypadku zmiennych rzeczywistych nasze dyskretyzacje mogą, ale nie muszą mieć jakiegokolwiek związku z właściwym podziałem na klasy. Jeśli zmienne nie są powiązane ze zmienną decyzyjną, to żaden podział na klasy nie ma znaczenia. Jeżeli jednak zmienne są powiązane ze zmienną decyzyjną, to tylko podział na klasy, który będzie zgodny ze sposobem powiązania ma sens - chcielibyśmy uzyskać takie przedziały, w których w każdym jedna z klas ma wysokie, a druga niskie prawdopodobieństwo. W ten sposób uzyskamy podział o niskiej entropii informacyjnej.
Przykład - załóżmy, że podzieliśmy studentów na dwie grupy - typowych i nietypowych pod względem atrybutów fizycznych (wzrost i waga). Typowy student, to taki, który nie odbiega za bardzo od średniej, nietypowy to będzie ktoś bardzo wysoki albo bardzo niski, podobnie z wagą. Żaden pojedynczy podział na dwie klasy nie jest w stanie dobrze rozdzielić studentów typowych od nietypowych, chociaż można uzyskać stosunkowo czystą jedną klasę. Możliwość uzyskania czystej klasy jest wskazówką, że zmienna może mieć znaczenie. Jednak szansa na uzyskanie takiego podziału losowo w jednej próbie nie jest duża. Dlatego należy przeprowadzić test wielokrotnie.
Można zauważyć również, że liczba podziałów może mieć bardzo duże znaczenie - w podanym przykładzie podzielenie każdej zmiennej na trzy przedziały da dużo lepszy efekt niż na dwa - w bardzo wielu losowych podziałach uzyskamy wynik, w którym przedziały skrajne mają duże prawdopodobieństwo jednej z klas a centralny drugiej.

Wybór właściwej liczby i właściwego sposobu generowania podziałów.

Wybór liczby podziałów jest wynikiem kompromisu między jak najwierniejszym odwzorowaniem przestrzeni a jak największą wiarygodnością wyniku. Szczegółowe odwzorowanie przestrzeni można uzyskać przez zwiększenie liczby podziałów. Jednak wraz ze wzrostem liczby podziałów spada liczba obiektów w przedziale. W szczególności gdy liczba podziałów jest bardzo duża, możemy doprowadzić do sytuacji, w której większość podziałów będzie zawierać jeden lub zero elementów. Wiarygodność oszacowania prawdopodobieństwa przy jednym obiekcie w przedziale jest bardzo niska.
W praktyce powinniśmy dążyć do tego, by średnia liczba obiektów w przedziale była w przybliżeniu równa pierwiastkowi z liczby wszystkich obiektów - wtedy optymalizujemy rozkład wariancji między przedziałami i wewnątrz przedziałów.

Czyli liczebność przedziału można oszacować jako: √N a optymalna liczba przedziałów w d-wymiarowym przypadku to $\sqrt[d]{\sqrt{N}}$.

W wypadku 1000 obiektów optymalna liczba przedziałów to około 30. Przy rozkładzie dwuwymiarowym optymalna liczba podziałów to 4-5 na wymiar (liczba przedziałów jest o jeden większa).

Przedziały powinny zawierać podobne liczebności obiektów, dlatego przedziały budujemy w następujący sposób - dla każdej zmiennej sortujemy obiekty według wielkości a następnie wybieramy losowo z rozkładu jednostajnego indeksy podziałów. Obiekty przypisujemy do klas zgodnie z tymi podziałami.

Zadanie:

Dany jest układ informacyjny składający się z N obiektów opisanych K zmiennymi opisowymi z dziedziny liczb rzeczywistych i jedną zmienną decyzyjną D przyjmującą wartości 0/1.

Napisz program, który znajdzie P par zmiennych, które są najbardziej powiązane ze zmienną decyzyjną. W tym celu przeprowadź wielokrotnie (256 razy) dyskretyzację zmiennych na odpowiednią liczbę przedziałów. Dla każdej dyskretyzacji znajdź wartość $GIG_{1->2}$. Znajdź maksymalne $GIG_{1->2}$ wśród wszystkich dyskretyzacji. Jeśli wartość $GIG_{1->2}$ jest większa niż poziom odcięcia to zapisz ją na listę kandydatów. Poziom odcięcia określamy przez przeprowadzenie próby na fragmencie danych po randomizacji (patrz wariant I zadania).
Dla ustalenia uwagi N = 1000, K = 30 000, P = 100 000, jednak algorytm powinien działać bez zmian dla N < 30 000.

Zasady oceniania:

Do zaliczenia zajęć należy dostarczyć poprawnie działającą implementację kartową jednej z wersji programu.
Program musi mieć dołączoną dokumentację, zawierającą opis zastosowanych podstawowych rozwiązań i optymalizacji.

Ocena będzie polegała na sprawdzeniu poprawności działania na zbiorze testowym (identycznym dla wszystkich, dostarczonym przez prowadzących), oraz sprawdzenie szybkości działania i porównanie jej z rozwiązaniem referencyjnym.

Każdy program działający poprawnie i szybciej od rozwiązania referencyjnego zostanie oceniony na ocenę bardzo dobrą.

Niezależnie od powyższego autor(ka) najlepiej (najszybciej) działającej implementacji każdej z wersji uzyska ocenę bardzo dobrą.

Dla programów niespełniających warunku szybkości działania ocenie podlegać będzie również opis implementacji, uzasadnienie przyjętych decyzji projektowych, opis kroków optymalizacyjnych.

Jeżeli autor dostarczy rozwiązanie bonusowe, to ocenione będą oba rozwiązania i podstawą oceny będzie lepszy wynik.