Algorytm najbliższych sąsiadów - Sprawozdanie

Rafał Kolaska

1 Cel zadania

Cel zadania jakie sobie postawiłem polegał na implementacji algorytmu najbliższych sąsiadów, służącego do klasyfikacji podobnych obiektów, z wykorzystaniem standardu programowania równoległego - OpenMP. Ponadto zostały opracowane funkcje normalizujące dane metodą Gaussa oraz Minimum-maximum.

2 Opis algorytmów

2.1 Normalizacja min-max i standaryzacja rozkładem normalnym

Normalizacja służy do przekształcenia danych o wysokich wartościach do przedziału [0,1]. W ramach zadania zostały zaimplementowane dwa algorytmy normalizujące:

1. Minimum - maksimum:

$$x^* = \frac{x - min(X)}{max(X) - min(X)} \tag{1}$$

   gdzie: min i max - wartości minimalna i maksymalna zbioru X; X - zbiór danych poddanych normalizacji; x - element zbioru X;

2. Gaussa

$$x^* = \frac{x - mean(X)}{std(X)} \tag{2}$$

   gdzie: mean - wartość średnia zbioru X; std - wartość odchylenia standardowego zbioru X; x - element zbioru X;

2.2 Algorytm klasyfikujący - najbliższych sąsiadów

Algorytm uczenia maszynowego oparty o miarę odległości. Służy do podziału obiektów na klasy według reguły klasyfikującej, która w tym przepadku, polega

na znalezieniu odległości do N najbliższych sąsiadów obiektu klasyfikowanego, pomiarze odległości w jakiej się od niego znajdują i przydziale do grupy, której członkowie mają przewagę w głosowaniu. Samo głosowanie może odbyć się przy różnych warunkach: większościowych lub wagowych. Również odległość może być mierzona według różnych dostępnych metryk. Najważniejszym parametrem charakteryzującym algorytm jest jego skuteczność, którą definiujemy jako:

$$S = skuteczno = \frac{liczba\ poprawnych\ klasyfikacji}{liczba\ wszystkich\ danych} \tag{3}$$

2.3 Metryki

W stosowaniu algorytmu można wykorzystać następujące miary odległości:

1. $\sqrt{\sum(x_i - y_i)}$ - odległość euklidesowa
1. $d = max|x_i - y_i|$ - odległość Czybyszewa
1. $d = \sum |x_i - y_i|$ - odległość manhattańska

2.4 Zrównoleglanie

W implementacji algorytmów posłużył język C++ wraz z interfejsem OpenMP, dzięki któremu zaostały zaimplementowane opcje zrównoleglania pętli. Głównymi charakterystykami uzyskanymi podczas procedowania algorytmu ML były: zależnosć czasu operacji od ilości wątków oraz przyśpieszenie definiowane jako stosunek czasu spędzonego nad zadaniem przez jeden wątek do czasu poświęconego przez liczbę wątków większą od 1:

$$a = \frac{T_1}{T_n} \tag{4}$$

Komputer wykorzystany w zadaniu: i5-7300HQ, RAM: 16GB

3 Opis danych

Dane pochodzą ze strony CIFAR-10 i zostały zebrane przez: Alex Krizhevsky, Vinod Nair i Geoffrey Hinton. Składają się z 5 plików danych testowych oraz 1 pliku danych referencyjnych. Każdy plik zawiera 10000 kolorowych obrazków o rozmiarze 32x32, w formie binarnej. Struktura w bajtach wygląda następująco: 1B - etykieta, 3072B - obraz. Reasumując, każdy obraz zajmuje 3072B ułożone w szeregu bez znaku końca linii. Etykiety opisują co przedstawia obraz. Wyrażone są przez liczby z zakresu od 0-9, gdzie każda liczba odpowiada pewnej kategorii. Obraz składa się z 1024 pikseli, a każdy z nich zawiera 3 wartości kolorów w formacie RGB. Etykiety: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

4 Wyniki

4.1 Wyniki pracy algorytmu klasyfikującego

Poniżej w tabelach spisano wartości skuteczności dla poszczególnych metryk, różnych rodzajów normalizacji oraz głosowania większościowego i wagowego. Najlepszy wynik daje klasyfikacja ważona na danych znormalizowanych metodą min-max. Każda metryka przy owych warunkach daje wyniki lepsze o kilka punktów procentowych niż w pozostałych przypadkach. Zdecydowanie zastanawia znaczne pogorszenie rozpoznawalności obrazków w przypadku korzystania z danych normalizowanych metodą Gaussa.

S	Metryka
0,324	Manhatańska
0,296	Euklidesowa
0,105	Czybyszewa

Rysunek 1: Głosowanie większościowe, normalizacja min-max

S	Metryka
0,266	Manhatańska
0,245	Euklidesowa
0,107	Czybyszewa

Rysunek 2: Głosowanie większościowe, normalizacja Gaussa

S	Metryka
0,331	Manhatańska
0,306	Euklidesowa
0,11	Czybyszewa

Rysunek 3: Głosowanie ważone, normalizacja min-max

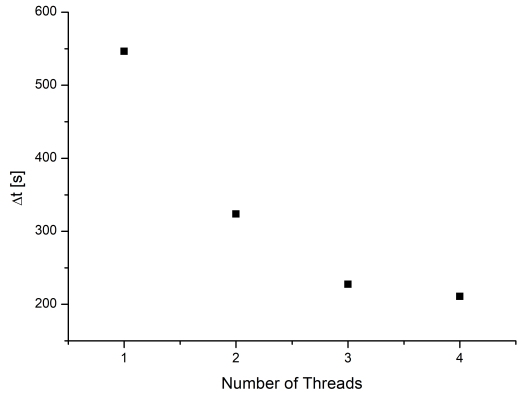
S	Metryka
0,236	Manhatańska
0,194	Euklidesowa
0,098	Czybyszewa

Rysunek 4: Głosowanie ważone, normalizacja Gaussa

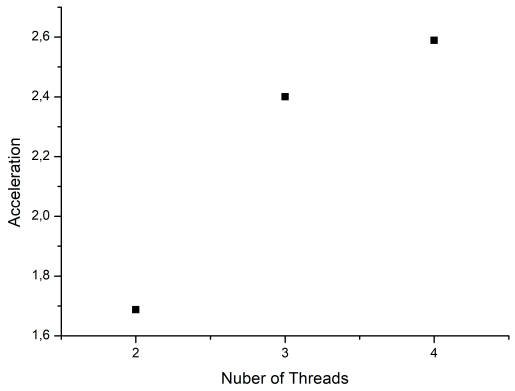
4.2 Wyniki zrównoleglania

Powstały dwa pliki z funkcjami normalizującymi metodami Gaussa i Minimum - Maksimum. W obu programach zostają zrównoleglone funkcje przetwarzające dane do postaci, którą można poddać normalizacji. W procesie klasyfikacji zrównoleglono pętlę, w której liczony jest dystans według wybranej metryki,

sortowane są odległości, a następnie analizowane pod kątem przydziału kategorii badanemu elementowi.
Poniżej zaprezentowane są wykresy zależności czasu od liczby wątków.



Rysunek 5: Zależność czasu od liczby wątków



Rysunek 6: Zależność przyśpieszenia od liczby wątków

Czas realizacji zadań maleje wykładniczo wraz ze wzrostem liczby wątków. Obliczenia wykonywane przez 4 wątki są ok. 2,5 raza szybsze niż przez pojedynczy wątek.