

Algorytmy Numeryczne – Zadanie 2

24 marca 2023

Macierze rzadkie

W praktyce zdarza się, że wiele elementów macierzy to zera, a stosunkowo niewiele elementów jest od zera różnych. Takie macierze nazywamy rzadkimi (ang. sparse). Dla macierzy rzadkich mamy zarówno możliwość optymalizacji użycia pamięci jak i czasu działania algorytmów.

Celem tego zadania jest implementacja metody rozwiązywania równań liniowych poznanej na algebrze i próba optymalizacji obliczeń przy założeniu, że macierz główna układu jest macierzą rzadką.

Struktury danych

Zdefiniuj klasę `MySparseMatrix` reprezentującą macierz nad ciałem liczb rzeczywistych. Elementy macierzy mają być zapamiętane tylko wtedy, gdy są różne od zera. Można to zrobić na wiele sposobów, przykładowe metody zapamiętania macierzy rzadkiej:

- DS1 słownik (ang. map), w którym kluczem jest para (r, c) opisująca numer wiersza i kolumny a wartością jest wartość elementu
- DS2 lista (lub tablica), w której każdy element opisuje jeden wiersz macierzy. Przy czym wiersz macierzy może być kolejną listą lub słownikiem
- DS3 lista (lub tablica), w której każdy element opisuje jedną kolumnę macierzy. Przy czym kolumna macierzy może być kolejną listą lub słownikiem
- DS4 inne wybrane, w tym połączenia metod powyższych

Zadanie

Zaimplementuj algorytm eliminacji Gaussa w następujących wariantach:

- A1 bez wyboru elementu podstawowego,
- A2 z częściowym wyborem elementu podstawowego,

używając przynajmniej dwóch różnych sposobów reprezentacji danych stworzonych dla macierzy rzadkich.

Implementacja algorytmów powinna być jedna (nie zależeć od użytej struktury danych). Do tego celu należy zdefiniować interfejs wspólny dla wszystkich wykorzystanych struktur danych.

Testy poprawnościowe

Rozwiąż układ równań liniowych

$$A \cdot X = B$$

Jako współczynniki macierzy A i wektora X proszę przyjąć pseudolosowe liczby zmiennoprzecinkowe z przedziału $[-1, 1)$ otrzymane jako iloraz $r/2^{16}$, gdzie r jest pseudolosową liczbą całkowitą z przedziału: $\{-2^{16}, 2^{16}-1\}$. Wektor B obliczamy jako $B := A \cdot X$. Wektor X zachowujemy jako rozwiązanie wzorcowe.

Macierz A i wektor B podajemy jako parametry do rozwiązywania układu równań. Przykładowo:

$$X' = \text{solve}(A, B)$$

Otrzymany wektor X' porównujemy z wektorem wzorcowym X .

Testy proszę przeprowadzić dla wielu macierzy różnych rozmiarów i przynajmniej trzech typów macierzy:

TG testy dla macierzy gęstych

TW testy dla macierzy wstęgowych (ang. band matrix).

TW testy dla rzadkich macierzy losowych, w których podobnie jak w macierzy wstęgowej elementy niezerowe znajdują się na głównej przekątnej a ponadto w wylosowanych miejscach poza główną przekątną.

Proszę zweryfikować następujące hipotezy:

H1 błędy dla metody A2 są mniejsze niż dla metody A1

H2 błędy rosną wraz ze wzrostem rozmiaru macierzy

Testy wydajnościowe

Proszę przetestować własną implementację porównując ją z wybraną istniejącą implementacją.

Proszę zweryfikować następujące hipotezy:

H3 czas działania zaimplementowanych algorytmów dla macierzy rzadkich jest zdecydowanie mniejszy niż dla macierzy gęstych tego samego rozmiaru.

H4 czas działania zaimplementowanych algorytmów rośnie z kwadratem rozmiaru macierzy jeśli przyjąć stałą liczbę niezerowych elementów w wierszu

Sprawozdanie

W sprawozdaniu proszę porównać otrzymane wyniki, sporządzić odpowiednie wykresy i sformułować wnioski na ich podstawie.

Praca zespołowa

Zadanie można wykonać w zespole dwu lub trzy osobowym. W takim przypadku proszę dokładnie oznaczyć jaki był zakres pracy członków zespołu. W oddaniu projektu musi uczestniczyć cały zespół.

Wskazówki

- Wszystkie testy proszę wykonać w taki sposób, aby można było je łatwo odtworzyć i powtórzyć (wykonywane eksperymenty można zachować w kodzie programu lub w skryptach uruchomieniowych, proszę zanotować wersję użytego oprogramowania, opcje kompilacji oraz konfigurację sprzętową).
- Macierze i wektory do testów należy losować raz dla różnych metod, w taki sposób aby porównywane operacje wykonywać na tych samych danych - jest to istotne zwłaszcza przy porównywaniu dokładności obliczeń.
- Można zerknąć na A Gentle Introduction to Sparse Matrices for Machine Learning – wprowadzenie w świat rzadkich macierzy przyjazne dla programistów Pythona – wraz z dalszymi odnośnikami do literatury.
- Proszę nie zwlekać do terminu oddania projektu