Московский авиационный институт
(государственный технический университет)

Факультет «Прикладная математика и физика»
Кафедра «Вычислительная математика и информатика»

Курсовой проект по

Языкам и методам программирования по теме:

«Разреженные матрицы»

Выполнил: Щербаков А.А.

Студент группы М8О-106Б

Преподаватель: Дубинин А.В.

Оценка:

Дата:

2017 год

Введение

С момента появления первых компьютеров их производительность заметно возросла. До сих пор количество транзисторов на микросхемах компьютера увеличивается согласно закону Мура – их количество удваивается каждые 2 года. Согласно другому наблюдению производительность также удваивается, но каждые 1,5 года. Этот факт неизменно остается действительным и в наши дни. Та же зависимость касается и хранилищ информации. Первые жесткие диски вмещали в себя всего несколько килобайт, а сейчас могут вместить вплоть до нескольких терабайт, и это еще не предел. Объем информации и повседневных задач также растут с каждым днем. Но потребность всегда растет быстрее, чем соответствующая возможность. Таким образом, хранить и обрабатывать некоторую информацию необходимо рационально. В пример можно привести разреженные матрицы. Они отличительны тем, что подавляющее количество элементов являются нулями. Это позволяет использовать некоторые ухищрения при их хранении и работе с ними.

Разреженные матрицы

Разреженная матрица — это матрица с преимущественно нулевыми элементами. В противном случае, если большая часть элементов матрицы ненулевые, матрица считается плотной. Единого мнения, какую матрицу считать разреженной, нет. Для матрицы порядка n число ненулевых элементов:

- есть O(n). Такое определение подходит разве что для теоретического анализа асимптотических свойств матричных алгоритмов
- в каждой строке не превышает 10 в типичном случае
- таково, что для данного алгоритма и вычислительной системы имеет смысл извлекать выгоду из наличия в ней нулей.

Огромные разрежённые матрицы часто возникают при решении таких задач, как дифференциальное уравнение в частных производных.

При хранении и преобразовании разрежённых матриц в компьютере бывает полезно, а часто и необходимо, использовать специальные алгоритмы и структуры данных, которые учитывают разрежённую структуру матрицы. Операции и алгоритмы, применяемые для работы с обычными, плотными матрицами, применительно к большим разрежённым матрицам работают относительно медленно и требуют значительных объёмов памяти. Однако разрежённые матрицы могут быть легко сжаты путём записи только своих ненулевых элементов, что снижает требования к компьютерной памяти. Таким образом, такие операции, как умножение матриц, вычисление определителя, решение дифференциальных уравнений могут быть выполнены гораздо быстрее при использовании более практичных алгоритмов.

Варианты размещения в памяти на языке Си

1. Цепочка ненулевых элементов в векторе A со строчным индексированием (индексы в массиве M равны -1, если соответствующая строка матрицы содержит только нули):

   В массиве M в ячейке с индексом i содержится индекс начала i-ой строки в матрице. В векторе A последовательно содержатся: номер столбца, значение данного элемента, индекс следующего ненулевого элемента этой строки или -1, если такого элемента нет. Если N – количество ненулевых элементов в матрице, а M – количество строк, то в худшем случае затраты по памяти будут O(2* (N+M)), то есть O(N+M). Вставка и удаление элементов в матрицу при этом будут весьма дороги: O(N).

2. Один вектор:

   Ненулевому элементу соответствуют две ячейки: первая содержит номер столбца, вторая содержит значение элемента. -1 в первой ячейке означает конец строки, а вторая ячейка содержит в этом случае номер следующей хранимой строки. -1 в обеих ячейках являются признаком конца перечня ненулевых элементов разреженной матрицы. Затраты памяти так же, как и в первом случае O(N+M), а вставка и удаление за O(N).

3. Три вектора:

   В первом векторе на i-ой позиции содержится индекс начала i-ой строки во втором и третьем векторе. Во втором векторе на i-ой позиции содержится номер столбца с ненулевым элементом, а в третьем векторе на той же позиции значение соответствующего элемента матрицы.

В данном случае затраты памяти O(2*N+M+1), то есть O(N+M). Вставка и удаление за O(N).

4. Два вектора:

В первом векторе содержится x - номер ненулевого элемента матрицы, который вычисляется по формуле:

x = i*M+j, где i, j – индексы ненулевого элемента в матрице, а M – количество строк. Для определения номера строки элемента по номе binary_search(tab->**key**, tab->**size**, x)

ру x подходит формула i = x/M, а номера столбца j = x%M, где / и %M - операции целочисленного деления и остатка по модулю M соответственно. Во втором векторе содержится ненулевое значение, содержащееся на x-ом месте в матрице.

Затраты памяти: O(2*N+1) = O(N). В отличие от остальных представлений добавление и удаление элемента происходит за O(1). Таким образом, данное представление одно из самых экономичных. Например, умножение транспонированного вектора на матрицу или матрицу на вектор может происходить за O(N), так как на результат влияют только ненулевые значения матрицы(и вектора).

Заключение

Проделанная мной работа показала, что несколько изменив типичное расположение информации в машинном представлении можно добиться значительного сокращения расходуемое памяти (O(N) вместо O(N^2) для разреженных матриц) и повышения быстродействия при работе с ней. Помимо этого, внешнее представление информации для пользователя остается неизменным, если раньше он перемножал две матрицы, то и сейчас он занимается тем же. Таким образом, изменение коснулось только внутреннего представления. Такой подход позволяет путем минимальных изменений уменьшить потребление программой памяти и ускорить ее работу. К примеру, при работе с большими матрицами порядка 10^8 многие алгоритмы становятся невыполнимыми за приемлемое время. Рассмотрим случай умножения разреженной матрицы на вектор. Обычный современный компьютер способен выполнять k*10^8 (1<=k<=9) операций за секунду. Используя алгоритм Винограда с некоторыми модификациями (O(N^2,3727)) умножение двух матриц порядка 10^8 займет (N^(8*2.3727)/N^8=10^11) секунд, что, несомненно, очень долго. Используя представление разреженной матрицы, мы добьемся выполнения такого умножения за 1 секунду. И это здорово.

Список источников

1.ru.wikipedia.org/wiki/Разреженная_матрица,«Теоретическая информация о разреженных матрицах»