

1 Разреженные матрицы

Попробуем определиться с понятием разреженная матрица. Выясняется, что это не так просто. Один из первых источников по данной тематике, в комплексе рассматривающий проблемы работы с разреженными матрицами, дает такое определение: разреженной называют матрицу, имеющую малый процент ненулевых элементов. В ряде других источников встречается такая формулировка: матрица размера $N \times N$ называется разреженной, если количество ее ненулевых элементов есть $O(N)$. Известны и другие определения.

Важно определиться с выбором структуры хранения – использовать обычные двумерные массивы или что-то более сложное, учитывающее факт разреженности. Именно в момент выбора структуры хранения вступают в дело перечисленные выше факторы. Так, в ряде случаев матрица имеет регулярную структуру (например, ленточная матрица), что позволяет разработать специализированную структуру хранения. Иногда размерность матрицы такова, что ее плотное представление попросту не укладывается в память. Если и разреженное, и плотное представление допустимы, а количество ненулевых элементов невелико, необходимо проанализировать, какая структура хранения будет более эффективна в рамках используемых алгоритмов и конкретной архитектуры. В зависимости от результатов этого анализа можно рассматривать матрицу либо как разреженную, либо как плотную.

В нашем случае (матрица размером $(N - 1)^3$ с семью ненулевыми элементами в строке) процент ненулевых элементов оценивается как $7/(N - 1)^3$, что при $N = 10$ уже можно считать много меньше 1. Так что вопросов не возникает - используем специальную упаковку для разреженных матриц.

2 Способы хранения матриц

Пусть N - количество элементов, NZ - количество ненулевых.

2.1 Координатный формат

Один из наиболее простых для понимания форматов хранения разреженных матриц – координатный формат. Элементы матрицы и ее структура хранятся в трех массивах, содержащих значения, их X и Y координаты (представлено на рис 1).

Плотное представление: $M = \text{const} * N^2$ байт (то есть представление массивом)
В координатном формате: $M = \text{const1} * NZ + \text{const2} * NZ + \text{const2} * NZ = O(NZ)$

Очевидно, что координатное представление по памяти гораздо круче.

В целом координатный формат достаточно прост, что является его несомненным достоинством, но оказывается недостаточно эффективным с точки зрения использования памяти и основных алгоритмов обработки.

A

1				2	
		3	4		
			8		5
	7	1			6

Структура хранения:

1	2	3	4	8	5	7	1	6
---	---	---	---	---	---	---	---	---

Value

0	0	1	1	3	3	5	5	5
---	---	---	---	---	---	---	---	---

Row

0	4	2	3	3	5	1	2	5
---	---	---	---	---	---	---	---	---

Col

Рис. 1: Покоординатное хранение

2.2 Формат CSR

Формат хранения, широко распространенный под названием CSR (Compressed Sparse Rows) или CRS (Compressed Row Storage), призван устранить некоторые недоработки координатного представления. Так, по-прежнему используются три массива. Первый массив хранит значения элементов построчно (строки рассматриваются по порядку сверху вниз), второй – номера столбцов для каждого элемента, а третий заменяет номера строк, используемые в координатном формате, на индекс начала каждой строки (представлено на рис 2)

A

1				2	
		3	4		
			8		5
	7	1			6

Структура хранения:

1	2	3	4	8	5	7	1	6
---	---	---	---	---	---	---	---	---

Value

0	4	2	3	3	5	1	2	5
---	---	---	---	---	---	---	---	---

Col

0	2	4	4	6	6	9
---	---	---	---	---	---	---

RowIndex

Рис. 2: Формат CSR

Встречаются разные виды такого хранения (например, индексация с нуля, хранение по столбцам и тд и тп)

Для начала мы будем работать с этим форматом хранения матриц.