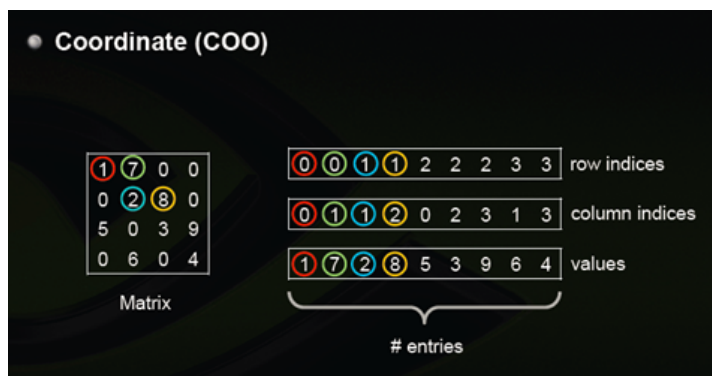


# Bin的专栏

## 稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB

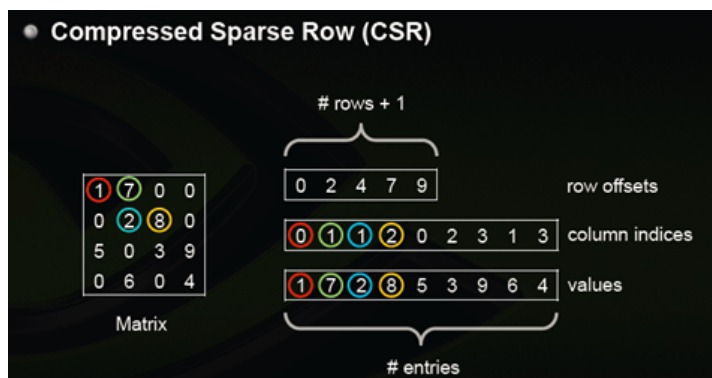
稀疏矩阵是指矩阵中的元素大部分是0的矩阵，事实上，实际问题中大规模矩阵基本上都是稀疏矩阵，很多稀疏度在90%甚至99%以上。因此我们需要有高效的稀疏矩阵存储格式。本文总结几种典型的格式：COO,CSR,DIA,ELL,HYB。

### (1) Coordinate (COO)



这是最简单的一种格式，每一个元素需要用三元组来表示，分别是（行号，列号，数值），对应上图右边的一列。这种方式简单，但是记录单信息多（行列），每个三元组自己可以定位，因此空间不是最优。

### (2) Compressed Sparse Row (CSR)



CSR是比较标准的一种，也需要三类数据来表达：数值，列号，以及行偏移。CSR不是三元组，而是整体的编码方式。数值和列号与COO一致，表示一个元素及其列号，行偏移表示某一行的第一个元素在values里面的起始偏移位置。如上图，第一行元素1是0偏移，第二行元素2是2偏移，第三行元素5是4偏移，第四行元素6是7偏移。在行偏移的最后补上矩阵总的元素个数，本例中是9。

CSC是和CSR相对应的一种方式，即按列压缩的意思。

以上图中矩阵为例：

Values: [1 5 7 2 6 8 3 9 4]

Row Indices: [0 0 1 3 1 2 2 3]

Column Offsets: [0 2 5 7 9]

### 公告

昵称: [Bin的专栏](#)  
园龄: [9年3个月](#)  
粉丝: [265](#)  
关注: [2](#)  
[+加关注](#)

### 导航

[博客园](#)  
[首页](#)  
[新随笔](#)  
[联系](#)  
[订阅](#)  
[管理](#)

<	2021年3月							>
日	一	二	三	四	五	六		
28	1	2	3	4	5	6		
7	8	9	10	11	12	13		
14	15	16	17	18	19	20		
21	22	23	24	25	26	27		
28	29	30	31	1	2	3		
4	5	6	7	8	9	10		

### 统计

随笔 - 29  
文章 - 0  
评论 - 63  
阅读 - 33万

### 搜索

### 常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

### 我的标签

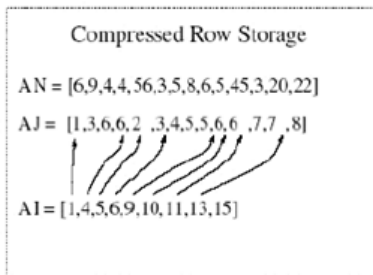
[Machine Learning\(14\)](#)  
[PRML\(6\)](#)  
[前沿动态-技术收录\(3\)](#)  
[论文阅读笔记\(2\)](#)  
[matlab\(2\)](#)  
[计算机科学\(1\)](#)  
[算法\(1\)](#)  
[信息检索技术\(1\)](#)

### 随笔分类

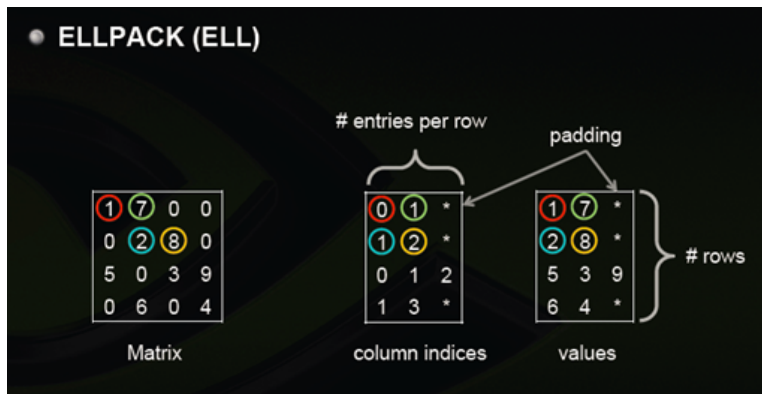
[All Articles\(29\)](#)  
[Matlab\(2\)](#)

再来看看一个CSR的例子[4]:

	1	2	3	4	5	6	7	8
1	6	9	4	5	3	20	22	
2								
3								
4								
5								
6								
7								
8								



### (3) ELLPACK (ELL)



用两个和原始矩阵相同行数的矩阵来存：第一个矩阵存的是列号，第二个矩阵存的是数值，行号就不存了，用自身所在的行来表示；这两个矩阵每一行都是从头开始放，如果没有元素了就用个标志比如\*结束。上图中中间矩阵有误，第三行应该是 0 2 3。

注：这样如果某一行很多元素，那么后面两个矩阵就会很胖，其他行结尾\*很多，浪费。可以存成数组，比如上面两个矩阵就是：

0 1 \* 1 2 \* 0 2 3 \* 1 3 \*

1 7 \* 2 8 \* 5 3 9 \* 6 4 \*

但是这样要取一行就比较不方便了

### (4) Diagonal (DIA)

论文阅读笔记(2)

前沿动态-知识收录(3)

算法、数据结构(2)

信息检索 Information Retrieval(2)

转载区(6)

### 随笔档案

2015年2月(5)

2015年1月(5)

2013年5月(2)

2013年4月(2)

2012年11月(3)

2012年9月(1)

2012年7月(3)

2012年3月(2)

2011年11月(6)

### 相册

Me(1)

### 最新评论

1. Re:随机采样方法整理与讲解 (MCMC、Gibbs Sampling等)

专门注册一下dd楼主

--clementin

2. Re:稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB

JDS格式、SELL格式、SELL-C-S格式、ESB格式，希望作者把这几种也补充进来，就更完美了。

--Fengqiao\_x

3. Re:稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB

good

--xiaoxi666

4. Re:机器学习降维算法一：PCA (Principal Component Analysis)

在矩阵的迹tr的那条公式那边，是否在min和tr之间少了一个负号 '-'，这样才能从最大化Max转变为Min?

--JeHuPeng

5. Re:随机采样方法整理与讲解 (MCMC、Gibbs Sampling等)

清晰易懂，厉害

--幻灭的星星

### 阅读排行榜

1. 随机采样方法整理与讲解 (MCMC、Gibbs Sampling等) (85821)

2. matlab绘图的坐标轴数字、范围、间隔控制 (59058)

3. 机器学习降维算法一：PCA (Principal Component Analysis)(32103)

4. 机器学习降维算法三：LLE (Locally Linear Embedding) 局部线性嵌入(28153)

5. 稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB(26615)

### 评论排行榜

1. 随机采样方法整理与讲解 (MCMC、Gibbs Sampling等) (12)

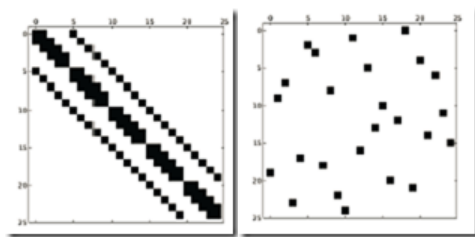
2. 机器学习-特征选择 Feature Selection 研究报告(7)

3. 今天开始学Pattern Recognition and Machine Learning (PRML)书，章

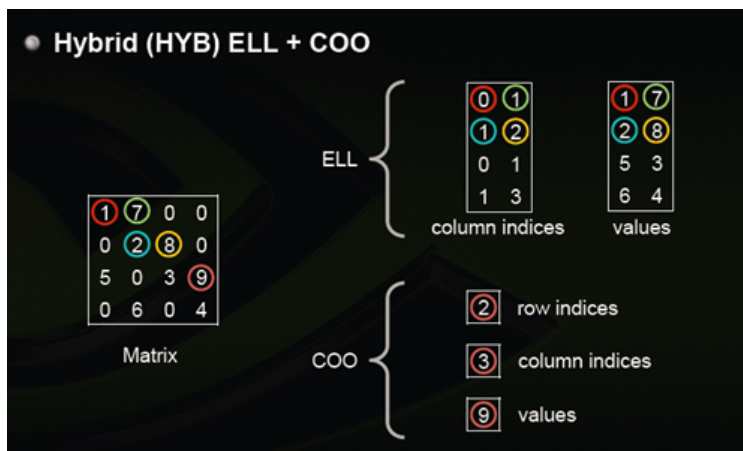


对角线存储法，按对角线方式存，列代表对角线，行代表行。省略全零的对角线。(从左下往右上开始：第一个对角线是零忽略，第二个对角线是5, 6，第三个对角线是零忽略，第四个对角线是1, 2, 3, 4，第五个对角线是7, 8, 9，第六第七个对角线忽略)。[3]

这里行对应行，所以5和6是分别在第三行第四行的，前面补上无效元素\*。如果对角线中间有0，存的时候也需要补0，所以如果原始矩阵就是一个对角性很好的矩阵那压缩率会非常高，比如下图，但是如果是随机的则效率会非常糟糕。



#### (5) Hybrid (HYB) ELL + COO



为了解决 (3) ELL中提到的，如果某一行特别多，造成其他行的浪费，那么把这些多出来的元素（比如第三行的9，其他每一行最大都是2个元素）用COO单独存储。

选择稀疏矩阵存储格式的一些经验[2]:

1. DIA和ELL格式在进行稀疏矩阵-向量乘积(sparse matrix-vector products)时效率最高，所以它们是应用迭代法(如共轭梯度法)解稀疏线性系统最快的格式;
2. COO和CSR格式比起DIA和ELL来，更加灵活，易于操作;
3. ELL的优点是快速，而COO优点是灵活，二者结合后的HYB格式是一种不错的稀疏矩阵表示格式;
4. 根据Nathan Bell的工作，CSR格式在存储稀疏矩阵时非零元素平均使用的字节数(Bytes per Nonzero Entry)最为稳定 (float类型约为8.5, double类型约为12.5)，而DIA格式存储数据的非零元素平均使用的字节数与矩阵类型有较大关系，适合于Structured Mesh结构的稀疏矩阵 (float类型约为4.05, double类型约为8.10)，对于Unstructured Mesh以及Random Matrix,DIA格式使用的字节数是CSR格式的十几倍;
5. 从我使用过的一些线性代数计算库来说，COO格式常用于从文件中进行稀疏矩阵的读写，如matrix market即采用COO格式，而CSR格式常用于读入数据后进行稀疏矩阵计算。

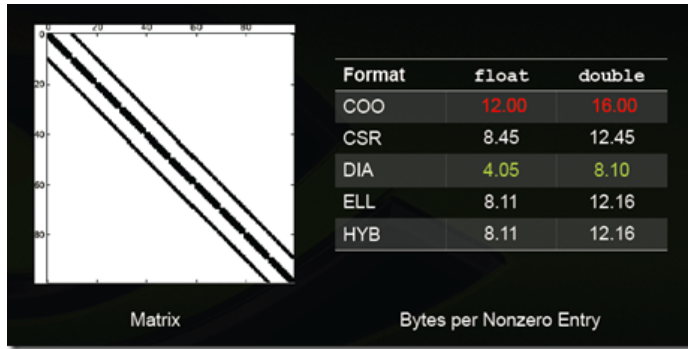
4. 机器学习降维算法三：LLE (Locally Linear Embedding) 局部线性嵌入(6)  
5. 距离计算方法总结(5)

#### 推荐排行榜

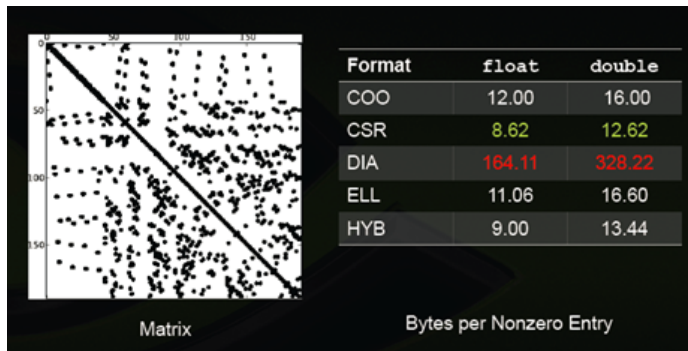
1. 随机采样方法整理与讲解 (MCM C、Gibbs Sampling等) (18)
2. 稀疏矩阵存储格式总结+存储效率对比:COO,CSR,DIA,ELL,HYB(9)
3. 机器学习距离公式总结(4)
4. 今天开始学Pattern Recognition and Machine Learning (PRML), 章节1.1, 介绍与多项式曲线拟合(Polynomial Curve Fitting)(4)
5. 机器学习-特征选择 Feature Selection 研究报告(4)

一些特殊类型矩阵的存储效率（数值越小说明压缩率越高，即存储效率越高）：

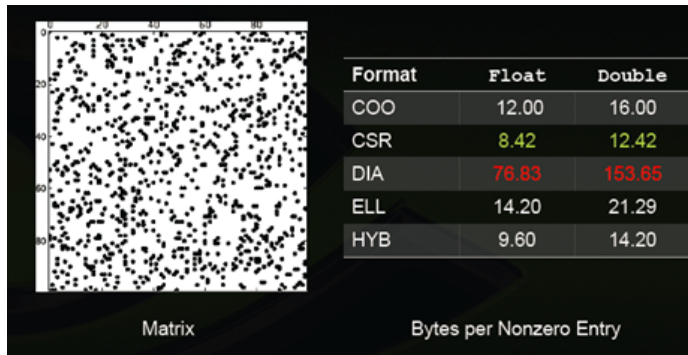
### Structured Mesh



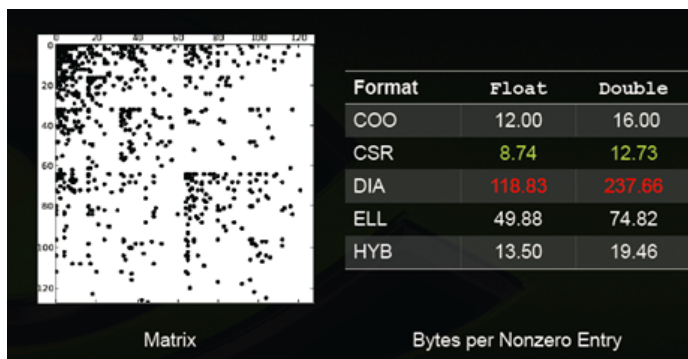
### Unstructured Mesh



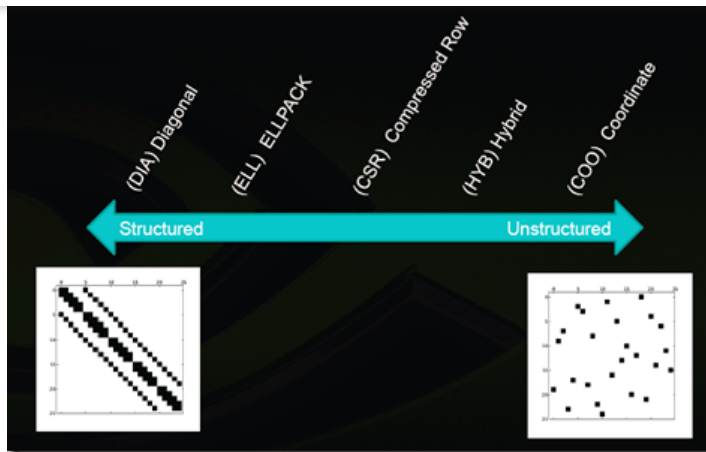
### Random matrix



### Power-Law Graph



格式适用性总结：



下面摘自[2]

## 6. Skyline Storage Format

The skyline storage format is important for the direct sparse solvers, and it is well suited for Cholesky or LU decomposition when no pivoting is required.

The skyline storage format accepted in Intel MKL can store only triangular matrix or triangular part of a matrix. This format is specified by two arrays: values and pointers. The following table describes these arrays:

values

A scalar array. For a lower triangular matrix it contains the set of elements from each row of the matrix starting from the first non-zero element to and including the diagonal element. For an upper triangular matrix it contains the set of elements from each column of the matrix starting with the first non-zero element down to and including the diagonal element. Encountered zero elements are included in the sets.

pointers

An integer array with dimension  $(m+1)$ , where  $m$  is the number of rows for lower triangle (columns for the upper triangle).  $\text{pointers}(i) - \text{pointers}(1) + 1$  gives the index of element in values that is first non-zero element in row (column)  $i$ . The value of  $\text{pointers}(m+1)$  is set to  $\text{nnz} + \text{pointers}(1)$ , where  $\text{nnz}$  is the number of elements in the array values.

## 7. Block Compressed Sparse Row Format (BSR)

The Intel MKL block compressed sparse row (BSR) format for sparse matrices is specified by four arrays: values, columns, pointerB, and pointerE. The following table describes these arrays.

values

A real array that contains the elements of the non-zero blocks of a sparse matrix. The elements are stored block-by-block in row-major order. A non-zero block is the block that contains at least one non-zero element. All elements of non-zero blocks are stored, even if some of them is equal to zero. Within each non-zero block elements are stored in column-major order in the case of one-based indexing, and in row-major order in the case of the zero-based indexing.

columns

Element  $i$  of the integer array columns is the number of the column in the block matrix that contains the  $i$ -th non-zero block.

pointerB

Element  $j$  of this integer array gives the index of the element in the columns array that is first non-zero block in a row  $j$  of the block matrix.

pointerE

[1] Sparse Matrix Representations & Iterative Solvers, Lesson 1 by Nathan Bell.

<http://www.bu.edu/pasi/files/2011/01/NathanBell1-10-1000.pdf>

[2] <http://blog.csdn.net/anshan1984/article/details/8580952>

[3] <http://zhangjunhd.github.io/2014/09/29/sparse-matrix.html>

[4] [http://www.360doc.com/content/09/0204/17/96202\\_2458312.shtml](http://www.360doc.com/content/09/0204/17/96202_2458312.shtml)

[5] Implementing Sparse Matrix-Vector Multiplication on Throughput-Oriented Processors, Nathan Bell and Michael Garland, Proceedings of Supercomputing '09

[6] Efficient Sparse Matrix-Vector Multiplication on CUDA, Nathan Bell and Michael Garland, NVIDIA Technical Report NVR-2008-004, December 2008

分类: [All Articles](#), [前沿动态-知识收录](#), [算法、数据结构](#)

标签: [前沿动态-技术收录](#), [算法](#)

好文要顶

关注我

收藏该文



Bin的专栏

关注 - 2

粉丝 - 265

[+加关注](#)

9

0

« 上一篇: [机器学习距离公式总结](#)

» 下一篇: [\[转载\]从机器学习谈起](#)

posted on 2015-02-04 23:05 [Bin的专栏](#) 阅读(26618) 评论(3) [编辑](#) [收藏](#)

## 评论

#1楼 2015-02-05 08:38 [Imageshop](#)

写的很好。

[支持\(0\)](#) [反对\(0\)](#)

#2楼 2018-05-02 21:02 [xiaoxi666](#)

good

[支持\(0\)](#) [反对\(0\)](#)

#3楼 2019-07-09 20:33 [Fengqiao\\_x](#)

JDS格式、SELL格式、SELL-C-S格式、ESB格式，希望作者把这几种也补充进来，就更完美了。

[支持\(2\)](#) [反对\(0\)](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能发表评论，立即 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)

## AWS免费产品:

- [如何在AWS上免费构建网站](#)
- [AWS免费云存储解决方案](#)
- [在AWS上免费构建数据库](#)
- [AWS上的免费机器学习](#)

## 最新新闻:

- [科幻方向盘被骗了！马斯克称目前看到的新款Model S并非最终版本](#)
- [百度递交香港上市文件：去年营收1071亿元 李彦宏投票权57%](#)
- [量子计算公司IonQ与SPAC dMY科技集团合并](#)
- [科学家重新编程细菌 以制造用于药物的设计分子](#)
- [被公众寄予厚望的日本超级计算机"富岳"开始全面运行](#)
- » [更多新闻...](#)