Optimizing Multiplications of Sparse Matrices

via

Size Estimation

Edith Cohen
AT&T Research

Sparsity and Prediction of Nonzero Structure

Many matrix operations are easier for sparse matrices.
Hence, sparse ouput matrix and sub-goals are desirable.

Example:
minimize fill-in in Gaussian eliminations.

<u>Matrix Multiplication:</u>

it is desirable to determine the nonzero structure of a matrix product "in advance":

- Optimal multiplication order:
Given matrices $A_{n_1 \times n_2}$, $B_{n_2 \times n_3}$, $C_{n_3 \times n_4}$.
Determine the faster way to compute ABC.
$(AB)C$ or $A(BC)$?

- Memory allocation and data structure set up for the multiplication.

Outline

- Review of sparse multiplication
- Review of the estimation scheme of [c94], tighter bounds, and applications to multiplication costs and column size estimations.
- Experimental evaluation and results.

Storage of Sparse Matrices

Compact. Only locations and contents of nonzero entries are stored.

Example: **Row Wise** representation:

$$\begin{pmatrix} 4 & 1 & 2 & 0 \\ 0 & 8 & 0 & -2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 7 \end{pmatrix}$$

row 1 $\longrightarrow$ $\boxed{1,4}$ $\boxed{2,1}$ $\boxed{3,2}$
row 2 $\longrightarrow$ $\boxed{2,8}$ $\boxed{4,\text{-}2}$
row 3 $\longrightarrow$ $\boxed{2,3}$
row 4 $\longrightarrow$ $\boxed{3,1}$ $\boxed{4,7}$

(other sparse representations: column-wise, arbitrary list of nonzeros)

Sparse Multiplication algorithm

Matrices $A_{n_1 \times n_2}$, $B_{n_2 \times n_3}$ can be multiplied in

$$\sum_{i=1}^{n_2} (\#\text{nonzeros in } A_{\bullet i})(\#\text{nonzeros in } B_{i \bullet})$$

operations.
$\Rightarrow$ cost of mult. is determined by row/column sizes

Example:

$$\begin{pmatrix} 4 & 1 & 2 & 0 \\ 0 & 8 & 0 & -2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 7 \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 \\ 0 & 0 & -3 & 4 \\ 0 & 1 & 6 & 7 \end{pmatrix}$$

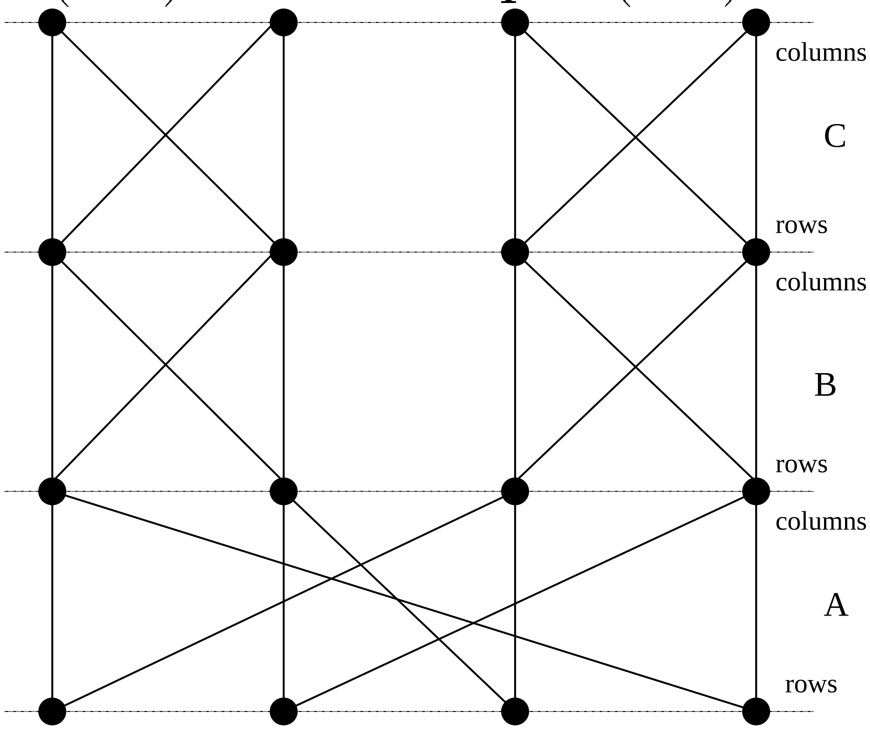
takes $1 \times 1 + 3 \times 2 + 2 \times 2 + 2 \times 3 = 17$ ops.

Optimal multiplication order

Example: $ABC =$

$$\begin{pmatrix} 4 & 0 & 2 & 0 \\ 0 & 8 & 0 & -2 \\ 0 & 1 & 7 & 0 \\ 2 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 5 & -2 & 0 & 0 \\ 9 & 1 & 0 & 0 \\ 0 & 0 & -3 & 4 \\ 0 & 0 & 6 & 7 \end{pmatrix} \begin{pmatrix} 6 & -1 & 0 & 0 \\ 4 & 7 & 0 & 0 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & -1 & 6 \end{pmatrix}$$

AB and BC take 16 operations.
The estimation alg. determines how many ops. needed for $A(BC)$and $(AB)C$:
$A(BC)$ takes 16 ops. $(AB)C$ takes 32 ops.



Nonzero Structure of Matrix Products

Consider the product $A = A_1 \cdots A_r$ ($r \geq 2$).
The **Non-Zero Structure** of A
(# of nonzeros in each row and column of A) is estimated in linear time in the number of nonzeros in $A_1, \ldots, A_r$. (considerably faster than MM time).
(Assuming no cancellation.)

The column/row-size estimates for matrix products can aid in:

- speeding up the multiplication and
- determining optimal multiplication order.

Asymptotic Behavior

- k-round estimation
- m NZ elements (total)
- S_i: number of NZ in column i of product.

For $1 \geq \epsilon > 0$ and $k > 0$:

$\diamond$ $O(km)$ time

$\diamond$ computes estimates $\hat{s_i}$ ($\forall$ columns i) s.t.:

  - For each column i,
    with probability $\geq 1 - e^{-\Omega(\epsilon^2 k)}$:
    $(1-\epsilon)S_i \leq \hat{s_i} \leq (1+\epsilon)S_i$

  - $Ex(|S_i - \hat{s}_i|) \leq S_i/\sqrt{k}$

$\diamond$ For $k = O(\epsilon^{-2} \log n)$,
with probability $1 - O(1/\operatorname{poly}(n))$, all estimates are within ϵ.

(see the paper for exact closed formula)

Estimating multiplication cost

Estimating the cost of multiplying $P_1 = A_1 \cdots A_i$ and $P_2 = A_{i+1} \cdots A_r$.

1. Estimate the column sizes $c_1, \ldots, c_n$ of P_1: $\hat{c}_1, \ldots, \hat{c}_n$.

2. Estimate the row sizes $r_1, \ldots, r_n$ of P_2: $\hat{r}_1, \ldots, \hat{r}_n$.

3. Estimate Mult. cost of $P_1 P_2$ by: $\Sigma_{i=1}^{n} \hat{c}_i \hat{r}_i$

Quality of cost-estimate:

⋄ No worse than for individual columns.

⋄ Usually better (errors "cancel out".)

• Estimate of the "optimal order" is found via dynamic programming.

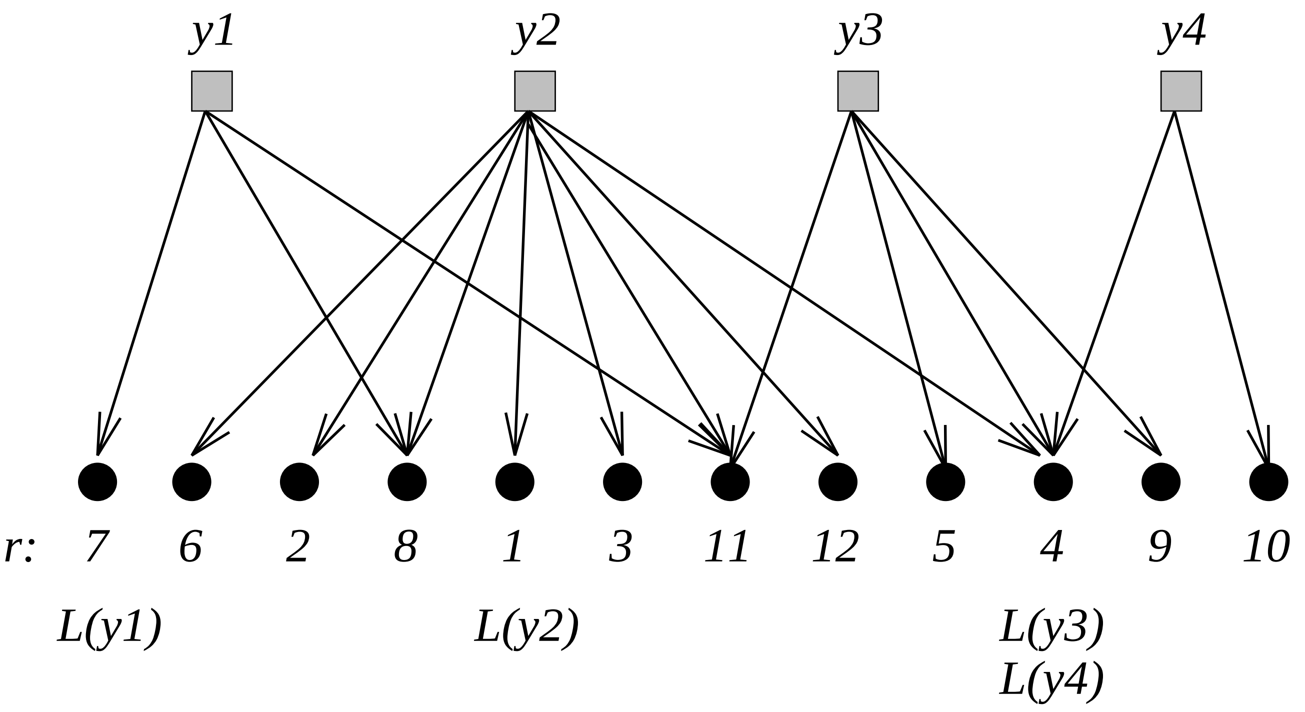
The estimation framework [C94]

- Sets Y and X
- $S : Y \to 2^X$ maps each $y \in Y$ to a subset of X

Goal: compute estimates $\hat{s}(y)$ of $|S(y)|$ $(\forall y \in Y)$

We have access to the following
Least-Element subroutine (LE):
Input: an ordering $r : X \to \{1, \ldots, |X|\}$ of X,
Output: a mapping $L : Y \to X$, such that:

- for all $y \in Y$, $L(y) \in S(y)$ and
- $r(L(y)) = \min_{w \in S(y)} r(w)$.

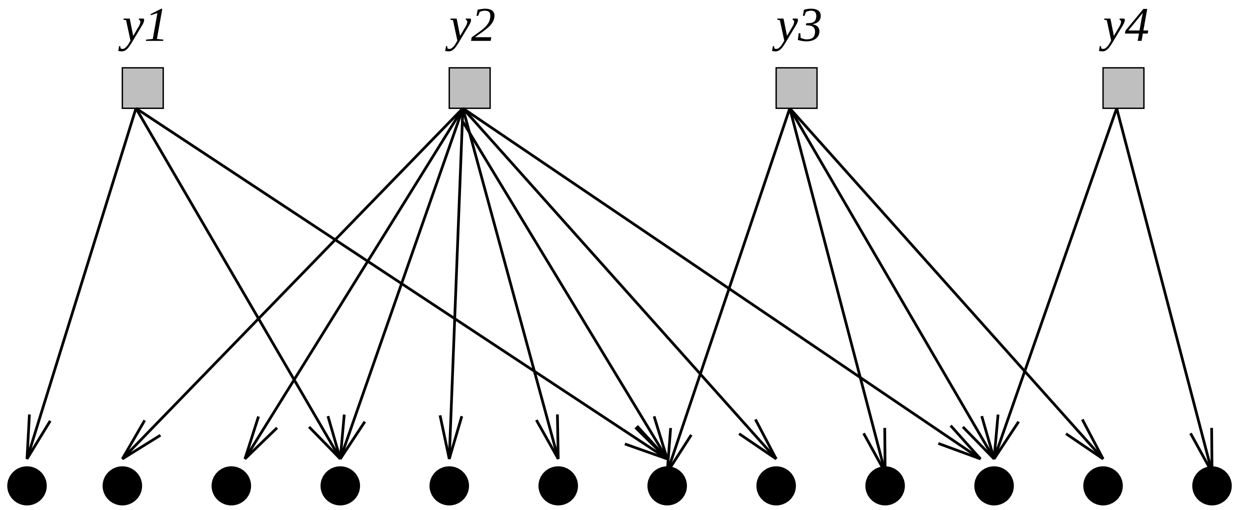


Intuition for use of LE to produce the estimates

- Select ranks $R : X \rightarrow [0,1]$ independently and uniformly at random

- Apply LE with the ordering induced by R

- $R(L(y))$ is the min of $|S(y)|$ values from $U[0,1]$. The expected value of $R(L(y))$ is $\frac{1}{|S(y)|+1}$ Hence, $|S(y)| = \frac{1}{Ex(R(L(y)))} - 1$



|S(y1)|=3 |S(y2)|=8 |S(y3)|=4 |S(y4)|=2

We expect *R(L(y4))* to be large and *R(L(y2))* to be small

The estimation algorithm

Repeat for k iterations $(1 \le i \le k)$:

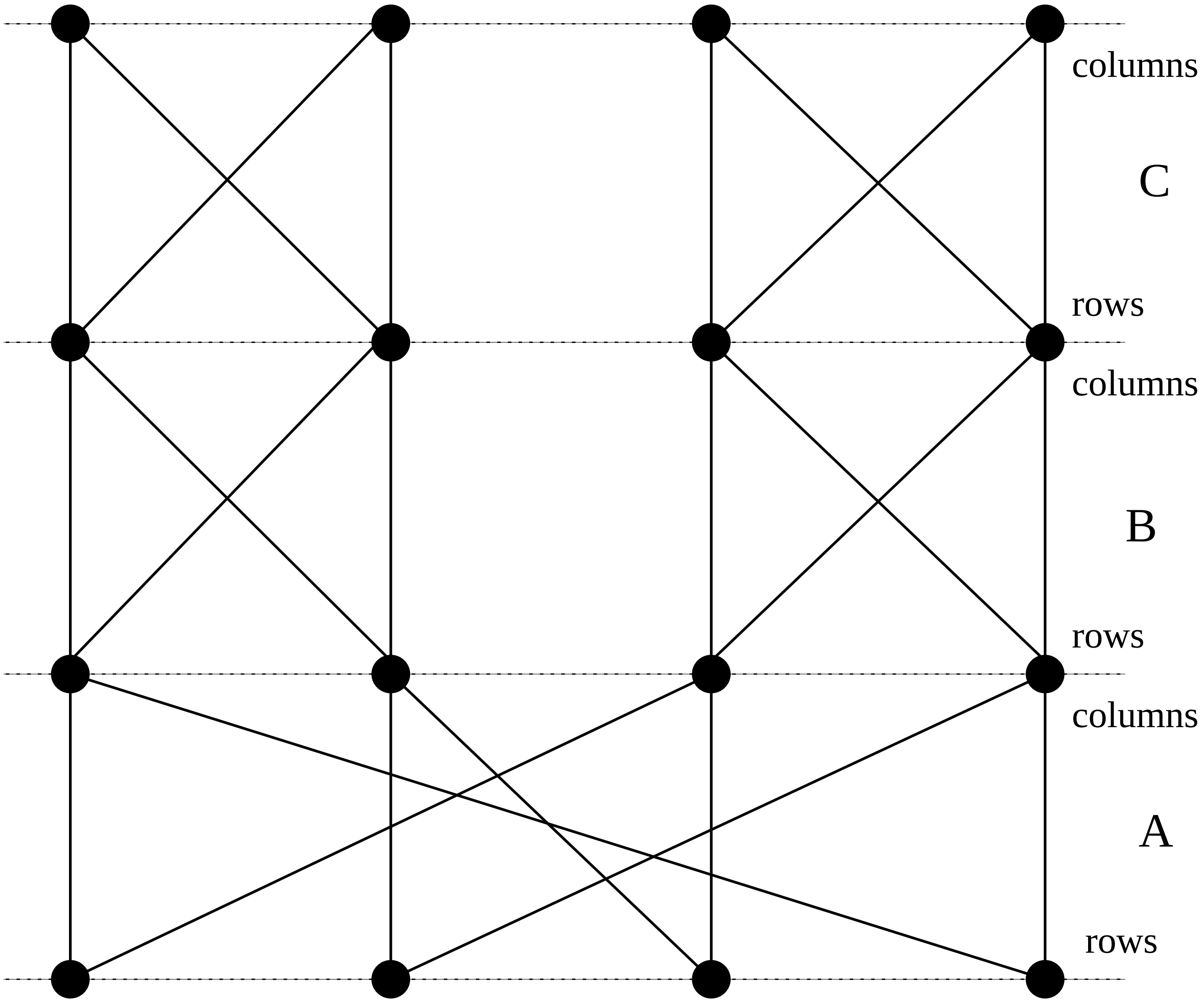
1. Select ranks $R_i : X \to [0,1]$, independently and uniformly at random.

2. Apply LE with the ordering induced by R_i. $L_i : Y \to X$ is the mapping returned by LE.

For each $y \in Y$:
$\hat{Ex}(y) \leftarrow \frac{\Sigma_{1 \le i \le k} R_i(L_i(y))}{k}$
(estimator for the expected value of $R(L(y))$)
$\hat{s}(y) \leftarrow \frac{1}{\hat{Ex}(y)} - 1$
(estimator for $|S(y)|$)

The quality of the estimates increases with k: For larger k (number of iterations), we get a better estimator for the expected minimum rank, and hence a better estimator for $|S(y)|$.

Estimating column sizes of AB



Experimental Evaluation

We experimentally evaluated the method for:

1. determining the least-costly order to compute ABC. (consider costs of estimation vs. multiplication and possible gain)

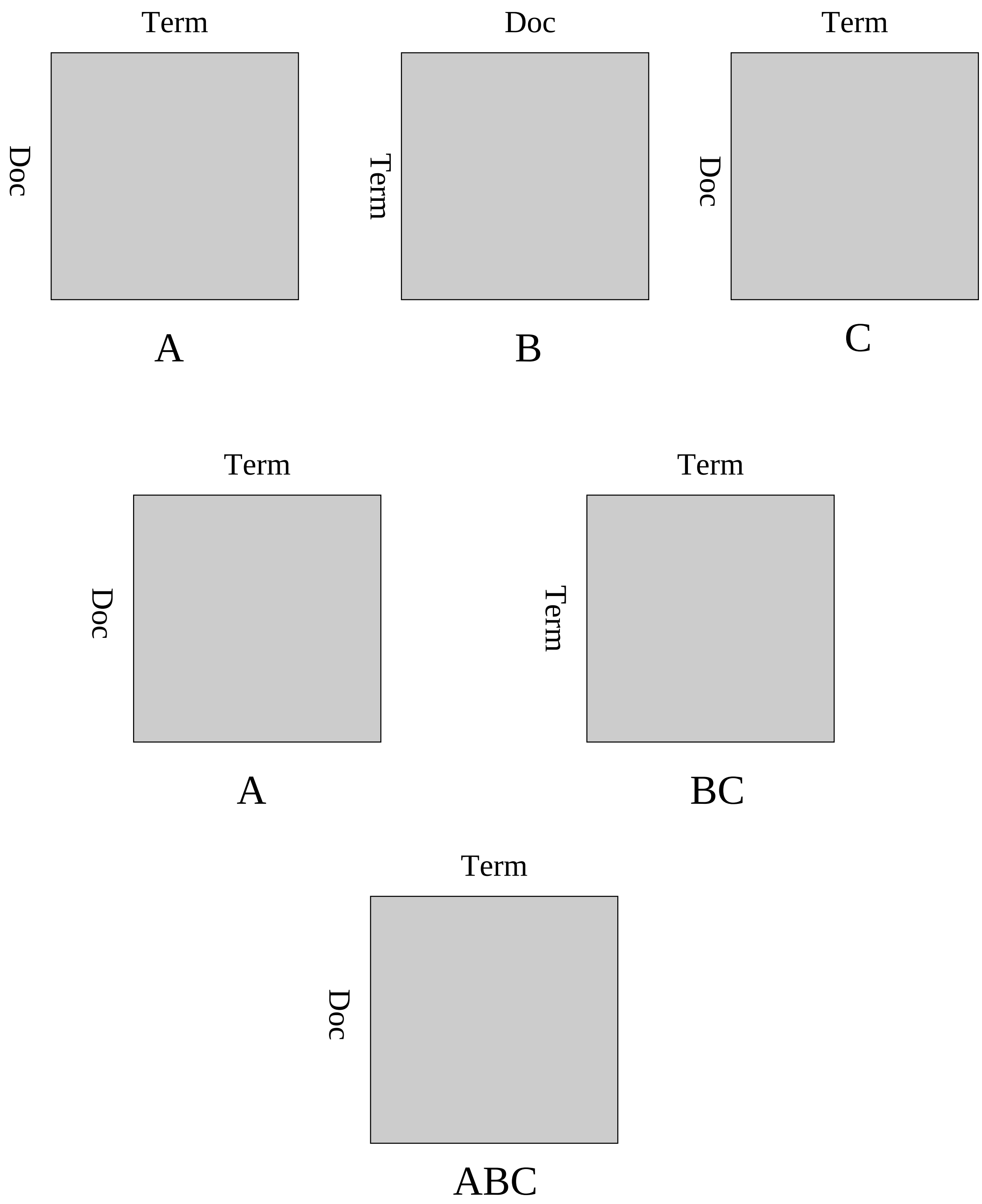
2. estimate individual column/row sizes in product.

If nonzero locations are "random", fast estimation methods that are based on densities can be deployed.
However, in many "real life" applications there are correlations between NZ locations.
In the experiments we used:

- generated "random" matrices that simulate likely correlations in NZ locations.

- some "real" IR data.

Example: Document-Term matrices



Simulating "real life" data

We used matrices A, B, C, such that rows of B and columns of C are "correlated.":

B, C: We select several block-types. every row/col have a random block-type. Probability of an entry to be nonzero is higher in the block than outside the block.

A: for every row we draw a random density (from a small set), every entry in the row is 0/NZ according to density.

Evaluation:

• We compare the number of operations for estimation and the number of operations for multiplications.
Note: "estimation" operation is a comparison. "mult." operation is float mult.

• We study the accuracy and dependence on number of rounds (vector size).

Visualizing matrices

Exp. E_0: Exp. E_1:

Our Experiments

Parameter choices:

Exp.	dimension	# blocks	block-size	p_{in}	p_{out}	A densities
E_0	256	8	15	0.8	0.02	{0.03}
E_1	512	8	60	0.25	0.01	{0.02}
1	1,000	20	50	0.3	0.005	{0.0150.0350.0200.0100.020}
2	1,000	10	30	0.4	0.005	{0.0150.0030.0200.0100.020}
3	1,000	10	60	0.2	0.005	{0.01}
4	5,000	100	40	0.5	0.0015	{0.0040.0040.0050.0050.004}
5	5,000	50	60	0.4	0.001	{0.0040.0030.0050.0050.003}
6	10,000	100	60	0.4	0.001	{0.0020.0010.0030.0030.002}
7	10,000	500	20	0.7	0.001	{0.002}

Matrix sizes:

Exp.	$\|A\|$	$\|B\|$	$\|C\|$	$\|AB\|$	$\|BC\|$	$\|ABC\|$	$c(A*B)$	$c(B*C)$
E_0	2,042	4,317	4,242	23,780	32,489	63,975	34,325	111,465
E_1	5,245	9,906	10,047	83,937	98,419	255,678	101,590	202,547
1	19,377	19,697	19,700	316,806	198,822	971,975	381,391	388,186
2	13,923	16,753	16,916	185,205	218,316	944,800	233,781	612,263
3	9,948	16,572	16,800	147,979	209,317	866,913	164,809	369,591
4	110,421	137,176	137,524	2,830,071	1,946,403	20,254,461	3,028,574	4,276,973
5	100,294	144,729	144,908	2,669,364	1,773,636	18,678,073	2,900,704	6,117,018
6	219,762	339,549	339,583	7,099,568	6,589,297	75,692,609	7,458,357	15,343,737
7	199,865	240,035	240,155	4,685,728	3,929,670	53,909,513	4,799,504	5,760,134
IR	82,670	85,793	85,155	5,787,707	1,174,978	12,305,303	8,870,366	2,412,967

Results

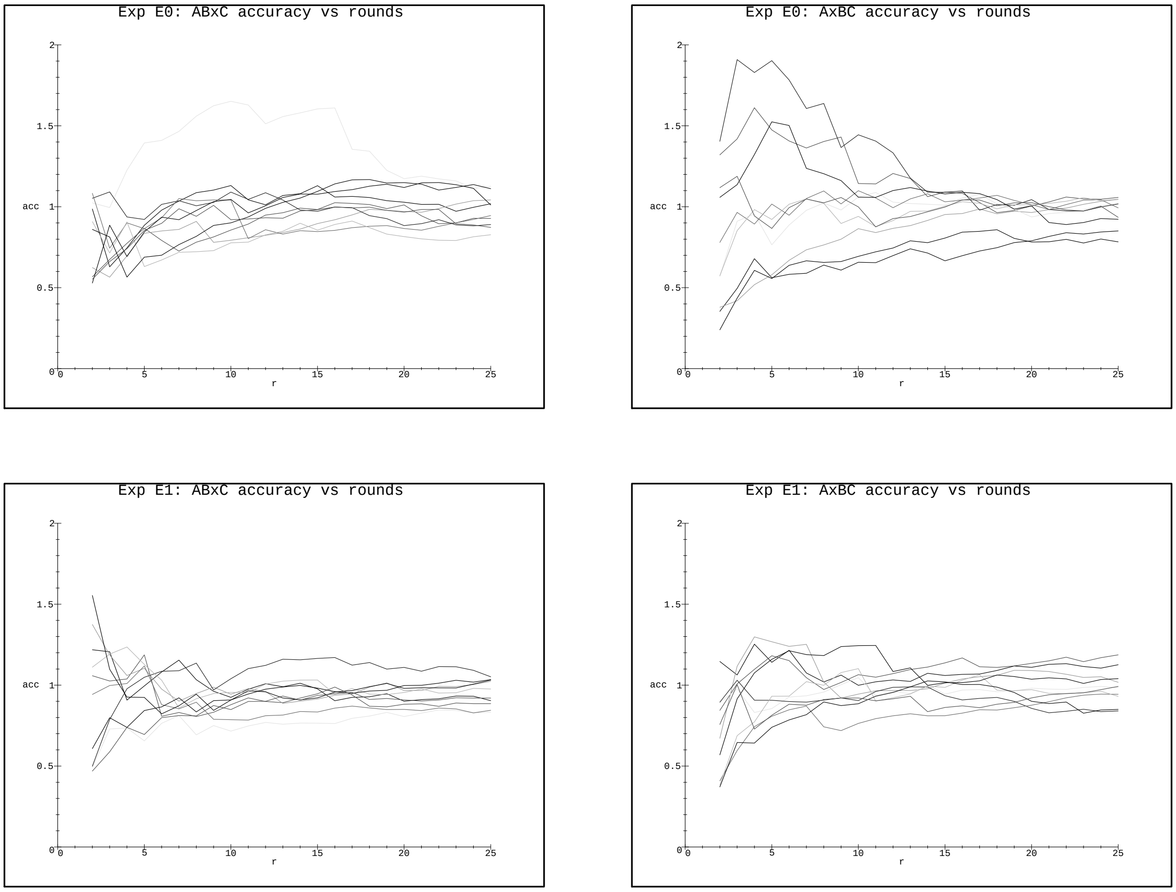
Exp.	c_r ($\times 10^3$)	cost: ($\times 10^3$)	exact	5 rnds	10 rnds.	15 rnds.	20 rnds.
E_0	14.9	$c(AB * C)$	582	382	465	520	489
		$c(A * BC)$	257	237	248	265	268
E_1	35.1	$c(AB * C)$	$1,709$	$1,887$	$1,613$	$1,563$	$1,652$
		$c(A * BC)$	$1,009$	822	929	870	930
1	78.5	$c(AB * C)$	$6,256$	$7,937$	$6,934$	$6,321$	$6,317$
		$c(A * BC)$	$3,860$	$5,168$	$4,163$	$4,235$	$4,246$
2	64.3	$c(AB * C)$	$6,456$	$7,760$	$7,416$	$6,465$	$6,647$
		$c(A * BC)$	$3,058$	$3,956$	$3,307$	$3,345$	$3,331$
3	59.9	$c(AB * C)$	$3,274$	$3,963$	$3,489$	$3,288$	$3,310$
		$c(A * BC)$	$2,084$	$2,690$	$2,222$	$2,287$	$2,286$
4	522.3	$c(AB * C)$	$88,001$	$93,130$	$95,477$	$88,911$	$83,709$
		$c(A * BC)$	$42,964$	$41,033$	$40,830$	$42,984$	$42,927$
5	534.7	$c(AB * C)$	$112,208$	$115,820$	$121,008$	$111,493$	$105,116$
		$c(A * BC)$	$35,510$	$33,486$	$33,106$	$35,031$	$352,821$
6	$1,238.4$	$c(AB * C)$	$319,546$	$298,115$	$306,531$	$313,904$	$322,776$
		$c(A * BC)$	$144,820$	$141,311$	$143,162$	$143,812$	$146,296$
7	920.1	$c(AB * C)$	$112,476$	$106,704$	$108,683$	$112,046$	$112,839$
		$c(A * BC)$	$78,607$	$74,261$	$76,961$	$76,949$	$78,113$
IR	339.4	$c(AB * C)$	$127,798$	$92,766$	$110,700$	$104,433$	$116,506$
		$c(A * BC)$	$68,892$	$45,018$	$50,595$	$58,875$	$65,785$

c_r: number of operations of a single estimation round

Acc. vs. Rnds for 10 reps. of Exp. E_0, E_1

Experiment E_0

Experiment E_1

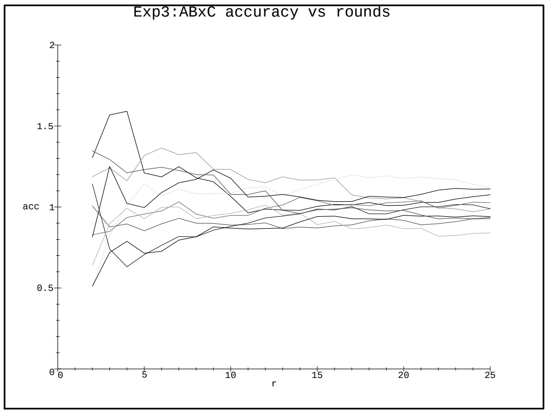


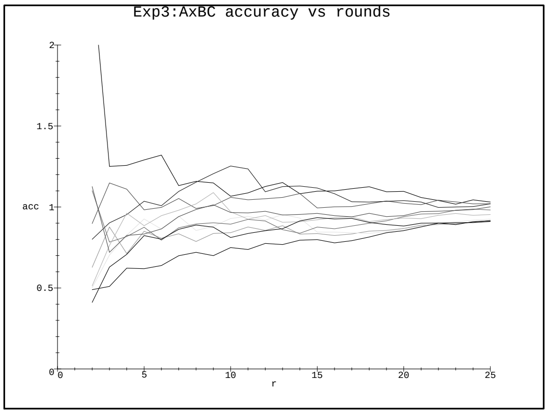
Cost of $(AB)C$

Cost of $A(BC)$

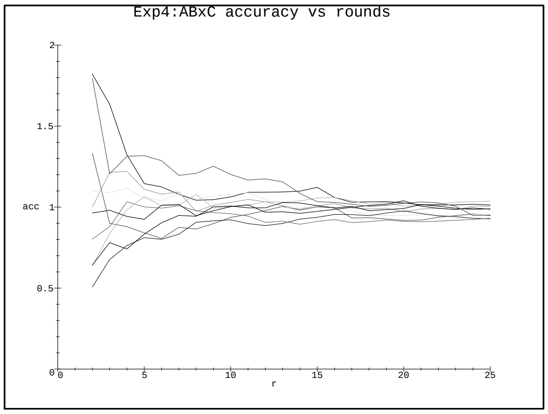
Convergence for Exp. 3,4,5

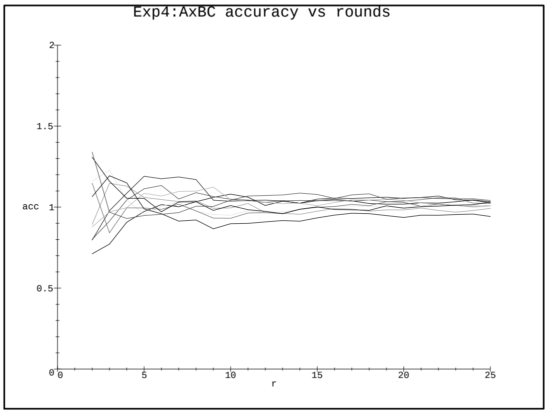
Experiment 3



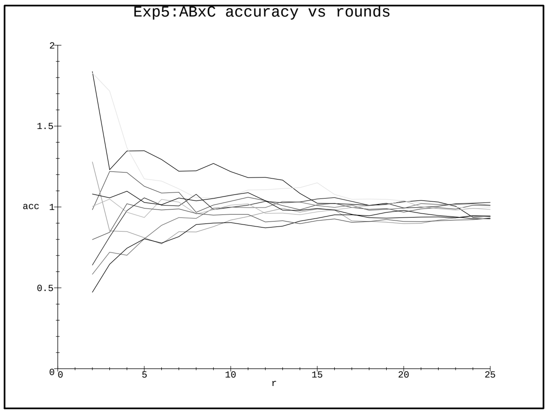


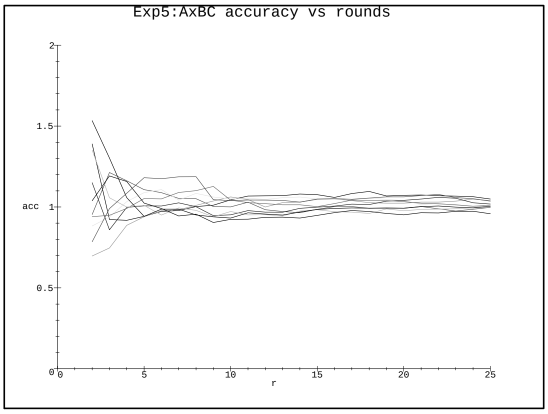
Experiment 4





Experiment 5



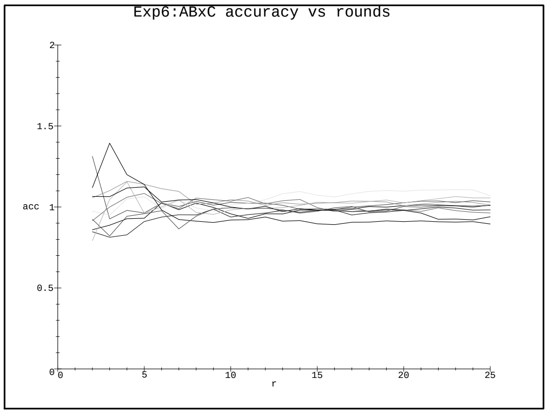


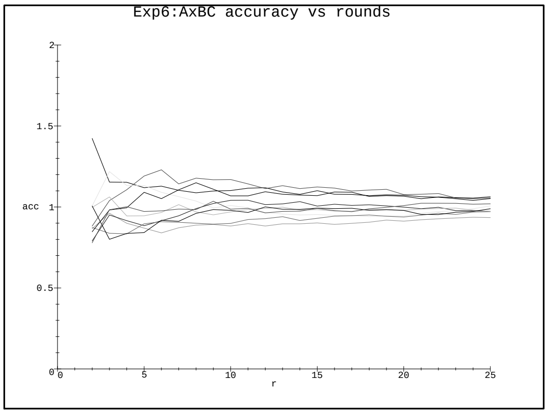
Cost of $(AB)C$

Cost of $A(BC)$

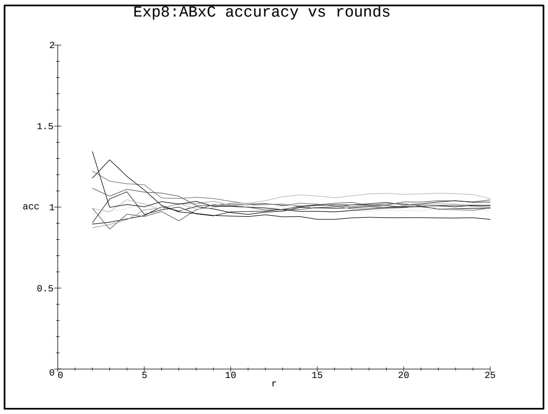
Convergence for Exp. 6, 7

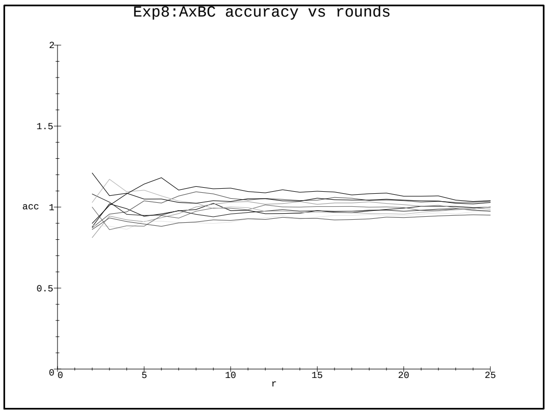
Experiment 6





Experiment 7



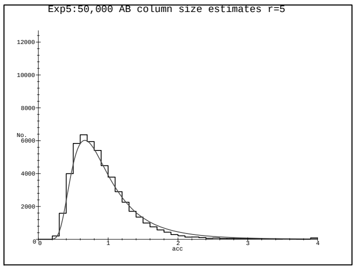


Cost of $(AB)C$

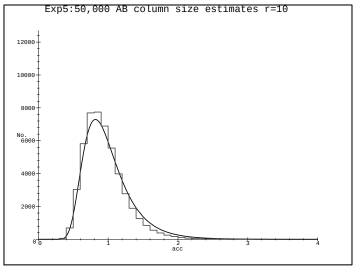
Cost of $A(BC)$

Acc. of estimates for individual AB column-sizes

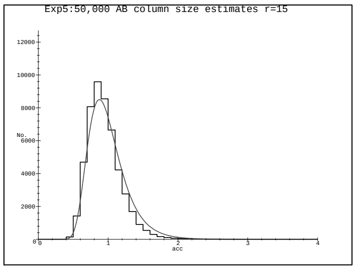
Histograms of ratio of estimated size to true size for $\{5, 10, 15, 20, 25\}$ rounds.



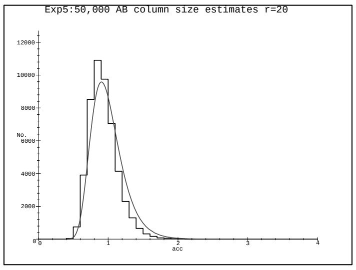
5 rounds



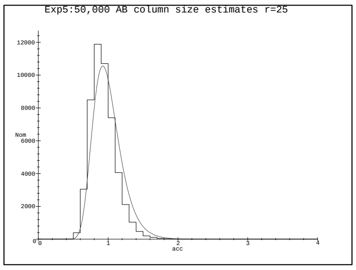
10 rounds



15 rounds



20 rounds



25 rounds

Conclusions from Experiments

- The number of rounds needed for good estimation decreases with matrix sizes.
- For Experiments 1-9 the estimation scheme was worth using. The cost of sufficiently good estimate of multiplication costs was small relatively for the gain by choosing the least expensive order.

  The matrices in Experiments 1–3 had dimension 10^3 and 10^4 nonzeros.