THE NUMBER OF BINARY $n \times m$ MATRICES WITH AT MOST k 1'S IN EACH ROW OR COLUMN

RICHARD J. MATHAR

Abstract. We count the the number of binary (0,1)-matrices with a given limit k on the number of 1's in each row and each column. The computation is recursive starting from the simplest case of the matrix with a single row.

1. Scope

Definition 1. *Let $A_{n,m,k}$ be the number of (0,1)-matrices with n rows, m columns and no more than k 1's in each of the rows and in each of the columns.*

Example 1. *The simplest example is*

$$A_{n,m,0} = 1, \tag{1}$$

because allowing no 1's in the matrices means only the matrix with all elements equal to 0 is admitted.

Example 2. *The number of binary matrices with a single column and no more than k 1's in that column is*

$$A_{n,1,k} = \sum_{f=0}^{k} \binom{n}{f}, \tag{2}$$

because the f 1's may be freely distributed over the column.

Example 3. *If the number of rows and the number of columns are both not larger than k, there is effectively no constraint on the placement of 1's:*

$$A_{n,m,k} = 2^{nm}, \quad n \le k \,\text{and}\, m \le k. \tag{3}$$

Summing its matrix elements down columns, each (0,1)-matrix can be categorized by a frequency vector with elements c_f counting the number of columns with f 1's, i.e., by the number c_0 of columns without any 1, the number c_1 of columns with one 1, and so on. The natural constraints for matrices restricted to k 1's are

$$\sum_{f=0}^{k} c_f = m, \tag{4}$$

$$0 \le c_f \le m, \quad \forall f. \tag{5}$$

Definition 2. *Let $A_{n,m,k}(c_0, c_1, \dots c_k)$ be the number of (0,1)-matrices with n rows, m columns, no more than k 1's in each row and each column, and with exactly c_f columns with f 1's.*

Date: November 23, 2014.
2010 *Mathematics Subject Classification.* 05A05, 15B35.
Key words and phrases. Combinatorics, Binary Matrices.

Example 4. *The* 2×3 *matrix*

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{6}$$

has two columns without 1, no column with one 1, and one column with two 1's, so $c_0 = 2$, $c_1 = 0$, $c_2 = 1$.

Example 5. *In a matrix with 1 row*

$$A_{1,m,k}(c_0, c_1, \ldots c_k) = \begin{cases} \binom{m}{c_1}, & c_1 \le k \text{ and } c_f = 0 \forall f > 1, \text{ and } c_0 + c_1 = m; \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

because we allow up to k *1's in that row and may distribute them over the* m *columns.*

Definition 3. *Let* $C(n,k)$ *denote the number of compositions of* n *into* k *non-negative parts.*

Remark 1. *The number* $C(n,k)$ *equals the number of compositions of* $n+k$ *into* k *positive parts. The Maple program in the Appendix generates the compositions into non-negative parts by (i) calling the function that generates positive parts and (ii) subtracting 1 from each of the parts.*

This frequency statistics refines the full count of matrices:

$$A_{n,m,k} = \sum_{C(m,k+1)=0^{c_0}1^{c_1}\cdots k^{c_k}} A_{n,m,k}(c_0, c_1, \ldots c_k). \tag{8}$$

Example 6. *The admitted matrices with* $m = 4$ *columns and up to* $k = 2$ *1's per column may separately by counted by the compositions* $4 = 4+0+0 = 3+1+0 = 2+2+0 = 1+3+0 = \cdots = 0+0+4$, *i.e. the matrices with 4 columns without 1's, the matrices with 3 columns without 1's and 1 column with one 1, the matrices with 2 columns without 1's and 2 columns with one 1 etc and eventually the matrices with 4 columns of two 1's.*

2. Recurrence

The number of admitted matrices with n rows is computed by considering the number of admitted matrices with $n-1$ rows and the number of ways of entering a total of up to k 1's in the final row distributed over the number of columns that have not yet exausted the upper limit of k in their count. The recurrence is anchored at Equation (7). We add a total of $N = d_0 + d_1 + \cdots + d_{k-1}$ 1's in the bottom row, d_0 of these placed at columns that had no 1's in the previous rows, d_1 placed at columns that hat one 1 in the previous rows and so on. The lower index of the d is limited to $k-1$ because we cannot insert 1's into columns that have already k 1's in the previous rows:

$$A_{n,m,k}(c_0 - d_0, c_1 - d_1 + d_0, c_2 - d_2 + d_1, \ldots, c_k + d_{k-1}) = \sum_{0 \le N \le k} \sum_{C(N,k)=0^{d_0}1^{d_1}\cdots(k-1)^{d_{k-1}}} \prod_{f=0}^{k-1} \binom{c_f}{d_f} A_{n-1,m,k}(c_0, c_1, \ldots, c_k). \tag{9}$$

The binomial factors on the right hand side count the number of ways of distributing d_f 1's in row n over the c_f columns that still admit additional 1's. The frequency vector on the left hand side shows that (i) adding d_0 1's to columns that had no

TABLE 1. The number $A_{n,m,1}$ of $n \times m$ binary matrices with at most one 1 in each row or column.

n	1	2	3	4	5	6	7	8	9
1	2								
2	3	7							
3	4	13	34						
4	5	21	73	209					
5	6	31	136	501	1546				
6	7	43	229	1045	4051	13327			
7	8	57	358	1961	9276	37633	130922		
8	9	73	529	3393	19081	93289	394353	1441729	
9	10	91	748	5509	36046	207775	1047376	4596553	17572114

TABLE 2. The number $A_{n,m,2}$ of $n \times m$ binary matrices with at most two 1's in each row or column.

n	1	2	3	4	5	6	7	8	9
1	2								
2	4	16							
3	7	49	265						
4	11	121	1081	7343					
5	16	256	3481	37441	304186				
6	22	484	9367	149311	1859926	17525812			
7	29	841	22009	490631	8871241	124920349	1336221251		
8	37	1369	46585	1386781	34589641	694936117	10876066069	129980132305	
9	46	2116	90811	3481543	114849676	3146625406	69238840861	1189279402021	15686404067098

1's in the previous rows diminishes the number of columns without 1's by d_0 and increases the number of columns with one 1 by d_0, that (ii) adding d_1 1's to columns that had a single 1 in the previous rows diminishes the number of columns with a single 1 by d_1 and increases the number of columns with two 1's by d_1, and so on.

Remark 2. *The implementation of (9) in the Maple program in the Appendix works with the reduced variables* $c'_0 \equiv c_0 - d_0$, $c'_f \equiv c_f - d_f + d_{f-1}$ *where* $1 \le f < k$ *and* $c'_k = c_k + d_{k-1}$.

3. RESULTS

The numbers $A_{n,m,k}$ are collected for $1 \le k \le 4$ in tables 1–4. Transposition does not effect the constraints on the maximum number of 1's, so these tables are symmetric $A_{n,m,k} = A_{m,n,k}$ and need only to be shown in the range $1 \le m \le n$.

On the diagonal of Table 1 we recognize the $A_{n,n,1}$ of [1, A002720]. The column $m = 1$ in Table 1 is a simple consequence of the fact that allowing a single 1 in a binary $n \times 1$ matrix allows either no one or allows that 1 in any of the n rows, $A_{n,1,1} = n + 1$.

On the diagonal of Table 2 we recognize the $A_{n,n,2}$ of [1, A197458]. The column $m = 1$ in that table shows [1, A000124] according to (2).

The column $m = 1$ in Table 3 shows [1, A000125] according to (2).

TABLE 3. The number $A_{n,m,3}$ of $n \times m$ binary matrices with at most three 1's in each row or column.

n	1	2	3	4	5	6	7	8
1	2							
2	4	16						
3	8	64	512					
4	15	225	3375	41503				
5	26	676	17576	386321	6474726			
6	42	1764	74088	2727835	79466726	1709852332		
7	64	4096	262144	15164605	724148776	26481406624	702998475376	
8	93	8649	804357	69214125	5103305401	300685003773	13310401771129	423669066884177
9	130	16900	2197000	268889923	29060188546	2608792241650	183396313726480	9574251908678125

TABLE 4. The number $A_{n,m,4}$ of $n \times m$ binary matrices with at most four 1's in each row or column.

n	1	2	3	4	5	6	7	
1	2							
2	4	16						
3	8	64	512					
4	16	256	4096	65536				
5	31	961	29791	923521	24997921			
6	57	3249	185193	10556001	532799101	21252557377		
7	99	9801	970299	96059601	8616972631	628094733099	34215495252681	
8	163	26569	4330747	705911761	106617548761	13564846995883	1332291787909909	944734
9	256	65536	16777216	4294967296	1037636664241	218509119324511	36998073025266151	4677696

$A_{n,n,\lfloor n/2 \rfloor}$ are in [1, A247158]. The hyperdiagonal $A_{n,n,n-1}$ yields [1, A048291], which means if a $n \times n$ matrix has at most $n-1$ 1's, there is at least one zero in each row or column, and flipping the elements of the matrices counts also the matrices with at least one 1 and therefore no fully blanked zero.

APPENDIX A. MAPLE IMPLEMENTATION

```
interface(quiet=true) :
# Compositions of n into k parts, each part >=0.
# @return A list of sublists, where each sublist is a composition of n and contains k nonnegative eleme
nonnCompo := proc(n::integer,k::integer)
        local c,co,e;
        # Empty list initially
        c := [] ;
        # Generate the compositions of n+k with k positive elements.
        # Generate the final list by subtracting 1 from each element.
        for co in combinat[composition](n+k,k) do
                [seq(e-1,e=co)] ;
                c := [op(c),%] ;
        end do:
        return c;
end proc:
```

```
# Number of n by m matrices with at most k 1's in each row and column
# @param n Number of rows
# @param m Number of columns
# @param k Upper limit of 1's in individual rows and columns
# @param freq freq[1] the number of columns with no 1. freq[i] the
#    number of columns with i-1 ones.
A := proc(n::integer,m::integer,k::integer,freq::list)
        local f,a,N,gr,contrib,transi,prefre,mu;
        option remember;
        # If the sum of the frequencies of 1's doesn't add up to the number
        # of columns (m), there is no such matrix.
        if add(f, f= freq) <> m then
                return 0 ;
        end if;
        # At most k 1's in each column, so the frequencies from 0 to k need to match.
        if nops(freq) <> k+1 then
                error "k is %d but freq has %d elements",k,nops(freq)
        end if;
        # If any frequency of 1's in a column is negative, there is no such matrix.
        for f in freq do
                if f < 0 then
                        return 0 ;
                end if;
        end do;
        if n = 1 then
                # Handle frequencies of a matrix with a single row.
                # 1 row, need only list[1]+list[2], others zero
                # Todo: this might be generalized to demand that freq(i)=0 for i>n+1.
                if nops(freq) > 2 then
                        if add(op(f,freq),f=3..nops(freq)) <> 0 then
                                return 0 ;
                        end if;
                end if ;
                # in the first row, the total number of 1's cannot be larger than k
                if nops(freq) > 1 then
                        if op(2,freq) > k then
                                return 0 ;
                        end if;
                end if;
                # list[1] the number of zeros out of m
                return binomial(m,op(1,freq)) ;
        else
                # sum up the number of matrices in a.
                a := 0 ;
                # recousre to A(n-1,m,k,freqprime)
                # add R=0: 1 way A(n-1,m,k,[c0,c1,c2..,ck-1]) -> A(n,m,k,[c0,c1,..ck-1])
                # add R=1: add 1 to c0' in binomial(c0',1) ways or add 1 to c1' in binomial(c1',1) ways
                # C(c0,1)*A(n-1,m,k,[c0,c1,..,ck-1]) + C(c1,1)*A(n-1,m,k,[c0,c1,...ck-1])+...
                # add R=2: add 2 to c0' in binomial(c0',2) ways or add 2 to c1' in binomial(c1',2) ways
                # or A(n,m,k,[c0,c1+2,...]
                # or mixed add 1 to c0' and 1 to c1' in binomial(c0',1)*binomial(c1',1)*A(n-1,m,k,[c0,c
                # A(n,m,k,[c0+1,c1+1,..]
```

```
                # C(c0,1)*A(n-1,m,k,[c0,c1,..,ck-1]) + C(c1,1)*A(n-1,m,k,[c0,c1,...ck-1])+...

                # N is the number of 1's in row n in the range 0<=N<=k.
                for N from 0 to k do
                        # That number of 1's can be split into gr[1] added to the columns with
                        # no ones yet, into gr[2] added to the columns with 1 ones yet,..
                        # added to the columns with k-1 ones yet. There cannot be 1's added to
                        # columns that already havy k ones, so this splitting of N is only
                        # into k groups, the last argument to nonnCompo.
                        for gr in nonnCompo(N,k) do
                                # last argument is not k+1, because we cannot add to freq[-1]
                                # The frequencies F[] of the previous matrix with n-1 rows undergo the
                                # F[0] -> F[0]-gr[0], F[1] -> F[1]+gr[0]-gr[1],.. F[k-1]->F[k-1]+gr[k-2
                                # F[k] -> F[k]+gr[k-1].
                                # Valid transitions demand that of course the F[i] are >=0, but
                                # (not to be overlooked) that all F[i]-gr[i] are also >=0, 0<=i<k.
                                # Now Maple indices are all 1 up:
                                # F[1] -> F[1]-gr[1], F[2] -> F[2]+gr[1]-gr[2],.. F[k]->F[k]+gr[k-1]-gr
                                # F[k+1] -> F[k+1]+gr[k] and all F[i]-gr[i]>=0, 1<=i<=k.
                                # And the frequencies f[] with the matrix of n rows
                                # are by solving to the right hand sides. f[1]=F[1]-gr[1], f[i]=F[i]-gr
                                # and f[k+1]=F[k+1]+gr[k].
                                # f[1]+gr[1] -> f[1]. f[2]+gr[2]-gr[1] -> f[2].... f[k]+gr[k]-gr[k-1]->
                                # f[k+1]-gr[k] -> f[k+1].
                                prefre := [op(1,freq)+op(1,gr),
                                                seq(op(f,freq)+op(f,gr)-op(f-1,gr),f=2..k),
                                                op(-1,freq)-op(-1,gr)] ;
                                transi := true;
                                for f from 1 to k do
                                         if op(f,prefre) < op(f,gr) then
                                                 transi := false;
                                                 break;
                                         end if;
                                end do:
                                if transi then
                                        mu  := mul( binomial(op(i,prefre),op(i,gr)), i=1..nops(gr) ) ;
                                        if mu > 0 then
                                                contrib := mu *procname(n-1,m,k,prefre) ;
                                                a := a+ contrib ;
                                        end if;
                                end if;
                        end do:
                end do:
                return a;
        end if;
end proc:

# n by m binary matrices with at most k 1's in each row or column
Agen := proc(n::integer,m::integer,k::integer)
        local a,freq;
        a := 0 ;
        # All possible combinations of sum(frequ)=m
        # freq[1]=c_0, freq[2]=c_1,... freq[k+1] = c_k
```

```
        for freq in nonnCompo(m,k+1) do
                a := a+ A(n,m,k,freq) ;
        end do:
        return a;
end proc:

# n by n binary matrices with at most k 1's in each row or column
Amain := proc(n::integer,k::integer)
        return Agen(n,n,k) ;
end proc:

A002720 := proc(n)
        Amain(n,1) ;
end proc:
seq(A002720(n),n=1..5) ;

A197458 := proc(n)
        Amain(n,2) ;
end proc:
seq(A197458(n),n=1..5) ;

A247158 := proc(n)
        Amain(n,floor(n/2)) ;
end proc:
seq(A247158(n),n=1..5) ;

Alatex := proc(k::integer)
        local n,m ;
        for n from 1 to 9 do
                printf("%d ",n) ;
                for m from 1 to n do
                        printf("& %d ", Agen(n,m,k)) ;
                end do:
                printf("\\\\\n") ;
        end do:
end proc:

Alatex(1) ;
Alatex(2) ;
Alatex(3) ;
Alatex(4) ;
```

References

1. Neil J. A. Sloane, *The On-Line Encyclopedia Of Integer Sequences*, Notices Am. Math. Soc. **50** (2003), no. 8, 912–915, http://oeis.org/. MR 1992789 (2004f:11151)

URL: http://www.mpia.de/~mathar

Max-Planck Institute of Astronomy, Königstuhl 17, 69117 Heidelberg, Germany