Mathematical Notes
2018-07-02

Constructing random invertible matrices over a finite field

Sebastià Xambó-Descamps and Narcís Sayols Baixeras

Abstract

The aim of this note is to describe an iterative construction of a random invertible matrix of any positive order over a finite field, to show that the disbribution of this matrix is uniform, and to describe a PyECC implementation of the algorithm. The algorithm is inspired in the D. Randall report [1].

Notations and conventions

$F = F_q$ will denote a finite field of cardinal q. The ring of F-matrices of order k will be denoted $F(k)$ and the multiplicative group of the $A \in F(k)$ that are invertible by $GL(k)$.

Given a random nonzero vector $v \in F^k$ we define the matrix I_v as the result of replacing the rth row of the identity matrix $I_k \in Gl(k)$ by v, where r is the index of the first non-zero component of v. It is clear that $I_v \in GL(k+1)$, as its determinant is equal to $v[r] \neq 0$.

1 Random extension of an invertible matrix

The main tool of this note consists of the following procedure:

1.1 (Procedure rd_extend(A))**.** Let $A \in GL(k)$.

1. Chose a random nonzero vector $v \in F^{k+1}$, and let r be the index of its first nonzero component.
2. Let $e_r \in F^{k+1}$ be the vector whose entries are all 0, except for a 1 in the component of index r, and define $A' \in GL(k+1)$ so that $A'[0] = e_r$, $A'_{0,r} = A$, and with random values $A'[j, r]$ drawn from F for $j = 1, \ldots, k$.
3. Return the matrix $B = A' I_v \in GL(k+1)$.

1.2 (Main argument)**.** Let G_v be the subset of the $B \in GL(k+1)$ whose first row is a given non-zero vector v. Then it is immediate that the map $G_{e_r} \to G_v$, $A' \mapsto A' I_v$, is bijective (the inverse map is given by $B \mapsto B I_v^{-1} = B I_{\bar{v}}$, where $\bar{v}_j = 0$ for $j < r$, $\bar{v}_r = 1/v_r$, and $\bar{v}_j = -v_j/v_r$ for $j < r$). This shows that $p(B) = p(v)p(A') = p(v)p(a)p(A)$, where $a = [a_{1r}, \ldots, a_{kr}]$, and the uniformity claim is clear because this value is a constant. Note that $p(v) = 1/(q^{k+1} - 1)$, $p(a) = 1/q^k$, and $p(A) = 1/N_k$, where N_k is the cardinal of $GL(k)$. If fact we have $p(B) = 1/N_{k+1}$, as it could not be otherwise, for it is easy to count that $N_k = (q^k - 1) \cdots (q^k - q^{k-1})$ and then $(q^{k+1} - 1)q^k N_k = N_{k+1}$.

1.3 (The function rd_insert(A,r))**.** This function implements the construction of A', once r is known:

```
def rd_insert(A,r):
    k = ncols(A)
    if r<0 or r>n: return "r has to be in 0..k"
    F = K_(A)
    A1 = matrix(F,k+1,k+1)
    A1[0,r] = 1
    for j in range(1,n+1):
        A1[j,r] = rd(F)
    if r==0:
        A1[1:,1:] = A[:,:]
    elif r==n:
        A1[1:,0:n] = A[:,:]
    else:
       A1[1:,0:r] = A[:,0:r]
       A1[1:,r+1:n+1] = A[:,r:n]
    return A1
```

1.4 (The function rd_extend(A))**.** Now we can write a function that implements the procedure rd_extend:

```
def rd_extend(A):
    k = ncols(A); F = K_(A)
    v = rd_nonzero_vector(F,k+1)
    r = 0
    for j in range(k+1):
        if v[j]!=0:
            r = j; break
    B = rd_insert(A,r)
    x = v[r]
    for j in range(r+1,k+1):
        B[:,j] = B[:,j]+ A[:,r]*v[j]
    B[:,r] = x*A[:,r]
    return B
```

2 The iterative procedure

Now the iterative procedure rd_GL(n,F) can be obtained with the following function:

```
def rd_GL(n,F=Zn(2)):
    a = rd_nonzero(F)
    A = matrix([[a]])
    for _ in range(2,n+1):
        A = rd_extend(A)
    return A
```

Note that the first step guarantees that a is a chosen uniformly at random among the non-zero elements of F. Then we iterate rd_extend ($n-1$ times), and the main argument shows that at the end we get, by induction, a matrix of $GL(n, F)$ chosen uniformly at random.

References

[1] D. Randall, "Efficient generation of random nonsingular matrices," 1991. Technical Report No. UCB/CSD-91-658, EECS Department, University of California, Berkeley.