Subject: Re: Random Sampling Without Replacement
Posted by Heinz Stege on Thu, 14 Oct 2010 01:14:00 GMT
View Forum Message <> Reply to Message

On Wed, 13 Oct 2010 09:46:09 -0600, David Fanning wrote:

> Folks,
>
> Has anyone coded up an IDL algorithm to do random
> sampling without replacement?
>
> For example, suppose I want to sample values in
> my 2D image. I want, say, 100 values that represent
> individual pixel locations in the image. How can
> I make sure I get 100 unique, but random, locations?
>
> Cheers,
>
> David

Hi all,

here is another way to do this calculation:

```
function unique_random,n,m
;
; n := total number of values
; m := number of samples
;
compile_opt defint32,strictarr,strictarrsubs
;
inds=long(randomu(seed,m)*(n-findgen(m)))
;
table=lindgen(n)
for i=0,m-1 do begin
  j=inds[i]
  inds[i]=table[j]
  table[j]=table[n-1-i]
end
;
return,inds
end
```

For a small number of samples (n=100000, m<50000) it is faster than Mike's code. And if the number of samples is not very small (n=100000, m>10000), it is even faster than JD's solution from http://tinyurl.com/26edmmq.

This is true in spite of the presence of the for-loop. I'm surprised myself. This algorithm may be a good over-all-solution for IDL.

Heinz