Subject: Re: Efficiently multiplying an array by a vector
Posted by Jeremy Bailin on Wed, 28 Sep 2016 15:42:04 GMT

View Forum Message <> Reply to Message

On Tuesday, September 27, 2016 at 8:38:15 PM UTC-5, LH wrote:

> The following text is a slightly edited version of a post by Mark Plonski several years ago that was never answered. Just saving typing time because my problem is the same. In fact, it seems like it must be a common problem when one tries to vectorize and speed up a program.
>
>
> What is the most efficient way to multiply every col in a 2D array
> (ncol x nrow) by a vector of length (ncol)?
>
> Example:
>
> input array   vector   output array
>
> a11 a12 a13   b1   a11b1 a12b1 a13b1
> a21 a22 a23   b2   a21b2 a22b2 a23b2
>
>
> This could be done by looping over the cols:
>
>   FOR i=0,ncol-1 DO c[i,*] = a[i,*] * b
>
> Is there a more efficient way (w.r.t. computational speed) to do this?
>
> I know I could replicate the column vector into a matrix (b # identity row)
> and then do a ptwise matrix multiply, but my matrices can
> be very large (1M elements) and I occasionally run out of swap
> space. I don't know if that would run any faster anyway.

I suspect that Wayne and Jeff's comments are the best option, but another idea is to do the matrix approach but break your matrix up into chunks to avoid memory problems. In a 2-chunk example:

```
b_matrix = b # indgen(nrow/2)
c[0,0] = a[*,0:nrow/2-1] * b_matrix
c[0,nrow/2] = a[*,nrow/2:*] * b_matrix
```

...and you can easily generalize that to an arbitrary number of chunks.

-Jeremy.