
Subject: Re: Transpose(A)*P*A

Posted by [Vince Hradil](#) on Fri, 10 Oct 2008 15:03:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Oct 10, 9:25 am, "mapper...@gmail.com" <zjwan...@gmail.com> wrote:

```
> hello,
>
> I have a question about how to improve the computation speed when deal
> with non-linear equation
>  $A \times x = I$ ,  $P$  is the weight for each
> row,  $P$  is  $M \times M$ 
>  $M \times N$   $N \times 1$   $M \times 1$ 
>
> then I have to build normal matrix which is
>  $\text{Transpose}(A) \times P \times A \times x = \text{Transpose}(A) \times P \times I$ 
>  $N \times N$   $N \times 1$   $N \times 1$ 
>
> then  $x$  can be solved.
>
> When  $M$  is bigger as 20000,  $N$  as 3000, the time to build  $AtPA$  is almost
> one hour, that is too long. the code is:
>
> PRO NormalMatrix, A, I, ATPA, ATPL, Weight = P
>   szA = SIZE(A, /DIMENSIONS)
>   if N_ELEMENTS(szA) ne 2 then begin
>     mess = dialog_message('Input matrix must be M*N format!', /
> information)
>     return
>   endif
>   szL = SIZE(L, /DIMENSIONS)
>   if (N_ELEMENTS(szL) ne 2 and szL[0] ne 1) then begin
>     mess = dialog_message('Input I must be 1*M format!', /information)
>     return
>   endif
>   if (szA[1] ne szL[1]) then begin
>     mess = dialog_message('Input A and I must be same rows!', /
> information)
>     return
>   endif
>   if not keyword_set(P) then P = fltarr(szA[1])*0.0+1.0
>   ATPA = fltarr([szA[0], szA[0]])
>   ATPL = fltarr(1, szA[0])
>
>   for i=0L, szA[0]-1 do begin
>     t1 = systime(1)
>     ATPA[i, i:szA[0]-1] = (p*A[i, *])##A[i:szA[0]-1, *]
>     ATPA[i:szA[0]-1, i] = ATPA[i, i:szA[0]-1]
>     ATPL[0, i] = (p*A[i, *])###
```

```
>         t2 = systime(1)
>         print, sza[0], i, t2-t1
>     endfor
>
> END
>
> I like to know how I can improve it to few seconds.
```

I think I'm confused - won't be the first time - why not just do the multiplication?

```
atpa = matrix_multiply(a,p##a,/atranspose)
```

or maybe it's

```
atpa = matrix_multiply(p##a,a,/btranspose)
```
