

# Statistics 243: *class notes*

William J. De Meo

October 8, 1997

## 1 Computations with Matrices

The operation  $X^t X$  is performed as  
`t(X) %*% X`  
in S. We will see how to do it faster.

### 1.1 Addition and Subtraction

Two matrices are conformable with respect to addition and subtraction, if they have the same number of rows and columns. If  $A$  and  $B$  are conformable, then

$$C = A \pm B \Rightarrow C(i, j) = A(i, j) \pm B(i, j)$$

as in the following C code segment:

```
double *a, *b, *c;
for (i=0; i<nrow *ncol; i++) c[i] = a[i]+b[i];
cnow=c;
for(i=0; i<nrow*ncol; i++)
*(c++) = a[i] + b[i]
```

Here's a skeleton for a matrix add function:

```
void madd(double *a, double *b, double *c, long n, long m){
    n*=m;
    while()
        (n--)*(c++) = *(a++) + *(b++);
}
```

### 1.2 Multiplication

Two matrices are conformable with respect to multiplication if their inner dimensions are the same; ie and  $m$  by  $n$  matrix is conformable with an  $n$  by  $p$  matrix.

$$C = A * B \Rightarrow C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$$

The dot product:

$$(x, y) = \sum_{i=1}^n x_i y_i$$

The  $i, j$  the element of  $C$  is the dot product of the  $i$ th row of  $A$  and the  $j$ th column of  $B$ . For example:

```

dot product for c[i][j]:
sum=0;
for(k=0;k<n;k++)
    sum+=a[i][k]*b[k][j];

```

```

1  6 11   1  4
2  7 12   2  5
3  8 13   3  6
4  9 14
5 10 15

```

Suppose all matrices are stored in column major format in a one dimensional array. Then we use a *stride* variable to indicate how many elements you must advance to get to the next entry in the same row. To change a col to a row or row to col (e.g when using the transpose) just need to change the stride from nrow to 1. So, the dot product routine would look like:

```

double dots(double *x, double *y, long n, long ix, long iy){
/* x is the address of x[i][0], y the address of y[0][j] */
/* ix and iy are the strides */
/* if ix = iy = 1, we have the ordinary dot product */

double sum = 0;
for(i=0;i<n; x+=ix, y+=iy, i++) sum += *x * *y;
return sum;

```

So for matrices the stride of the first matrix should be nrow, while the stride for second matrix is 1. But if you want to multiply the transpose of the first matrix and the second matrix we put the stride of the first matrix to 1. So we only need one routine to do  $AB$   $A'B$  and  $AB'$ .