

## C++线性代数库Armadillo

做**机器学习**的同学一定对Matlab矩阵运算的简洁高效映像深刻，并且已经习以为常这种便捷。但实际工作中，我们经常需要将matlab代码移植成C/C++。

C++下的线性代数库诸如lapack, openblas之类接口生涩，例如矩阵乘法接口，eigen库又显得过于庞大。因此我一直在寻找一个接口简洁，同时对windows友好的轻量级C++线性代数库，直到我发现了这个：

Amradillo C++线性代数库：<http://arma.sourceforge.NET/>



Amradillo的名字来自于"犰狳",一种浑身鳞甲的小动物，Amaradillo的底层计算依赖于Lapack, 高性能计算则可以指定MKL或者openblas支持，对于上层使用者而言，Amradillo的接口非常友好，例如你可以直接采用+\*/运算符来操作矩阵：

```
[cpp]
01. mat A = randu<mat>(5,10);
02. mat B = randu<mat>(5,10);
03. mat C = randu<mat>(10,5);
04.
05. mat P = A + B;
06. mat Q = A - B;
07. mat R = -B;
08. mat S = A / 123.0;
09. mat T = A % B;
10. mat U = A * C;
11.
12. // V is constructed without temporaries
13. mat V = A + B + A + B;
14.
15. imat AA = "1 2 3; 4 5 6; 7 8 9;";
16. imat BB = "3 2 1; 6 5 4; 9 8 7;";
17.
18. // compare elements
19. umat ZZ = (AA >= BB);
```

以下代码展示了一个求解线性方程组的调用过程(在matlab中，这相当于求解 $X=A\backslash B$ )

```
[cpp]
```

```
01. | mat A = randu<mat>(5,5);
02. | vec b = randu<vec>(5);
03. | mat B = randu<mat>(5,5);
04. |
05. | vec x = solve(A, b);
06. | mat X = solve(A, B);
07. |
08. | vec x2;
09. | bool status = solve(x2, A, b);
```



收藏到代码笔记

armadillo库一直在更新，截止这篇博文撰写时，最新的版本是5.20，发布时间是2015-5-24。

来源：<http://blog.csdn.net/visionhack/article/details/46580779>