

Julia

Tree-Structured Graphical Models

CRF

Julia (our code)

SVM-struct

C++ (the library) (svm-struct code)

MATLAB (our code)

(compute max violated constraint)

AGM (ours)

Julia (our code)



arg max

Veterbi algorithm on trees

Tree-Structured Graphical Models

Model Complexity

CRF < SVM-struct < AGM (ours)

Runtime (training – 20 splits)

CRF < AGM (ours) < SVM-struct

tens of minutes

few hours (1-2 h)

a day or more

Programming Languages

Speed

Convenience

	Compiled source codes are translated before runtime	Interpreted source codes are translated on the fly
Speed	Static types are checked before runtime C, C++ Fortran Go Rust Swift	
Convenience	Dynamic types are checked on the fly	Python Ruby, Perl MATLAB R Lua

Programming Languages & Linear Algebra

PL: e.g. Python, R, MATLAB

Linear Algebra library: e.g. NumPy, R matrix, MATLAB
(usually written in C++)

BLAS (Basic Linear Algebra Subprograms)
LAPACK (Linear Algebra Package)
(written in FORTRAN)
e.g. Intel MKL & OpenBLAS

That's why: "vectorized your code to make it fast"

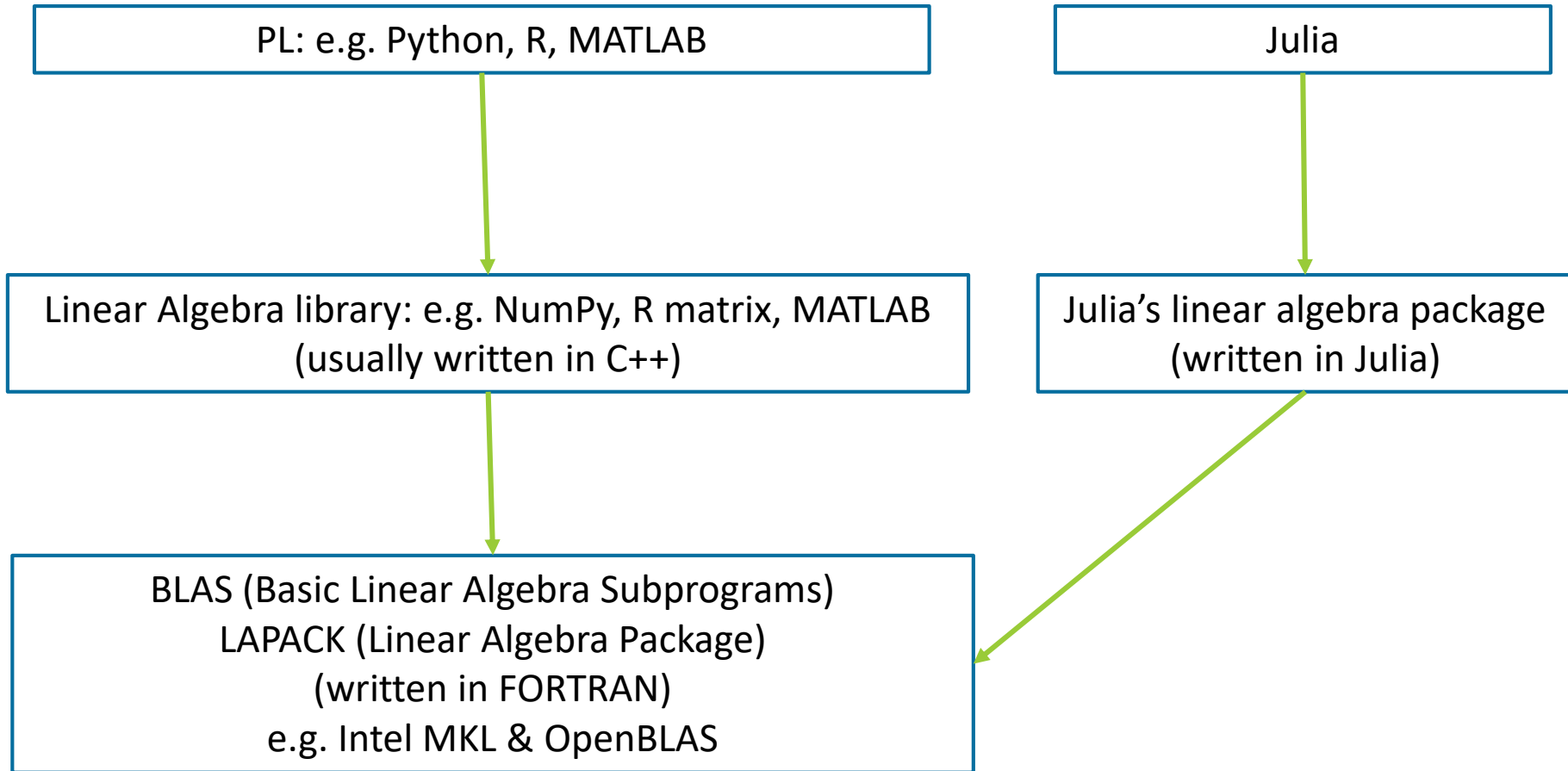
Programming Languages

Speed

Convenience

	Compiled source codes are translated before runtime	Interpreted source codes are translated on the fly
Speed	Static types are checked before runtime C, C++ Fortran Go Rust Swift	
Convenience	Dynamic types are checked on the fly Julia	Python Ruby Perl MATLAB R

Programming Languages & Linear Algebra



That's why: "vectorized your code to make it fast"

Some issues

- New language -> not many packages

It's very easy to load Python's package

- No fully usable debugger

Debugger is in ongoing work.

ASTInterpreter2 provides basic debugging

- A relatively more complex language

easy to learn, takes time to master?

Programming Languages

Speed

Convenience

	Compiled source codes are translated before runtime	Interpreted source codes are translated on the fly
Speed	Static types are checked before runtime C, C++ Fortran Go Rust Swift	
Convenience	Dynamic types are checked on the fly Julia MATLAB (>= 2015b) Python + Numba Lua + JIT	Python Ruby Perl MATLAB R

Conclusion

- MATLAB / Python / Julia

Good performance if you can vectorize (/tensorize) your code

- Julia

Good performance in any cases (if written properly)