# PPL: Pomegranate

“pomegranate was designed to be easy to use while not sacrificing on computational efficiency.”

Instalação e utilização:

A instalação é bastante simples. Pomegranate é, na verdade, um package de python. Portanto, basta seguir estes passos:

1. Instalar o Python;
2. Instalar o PIP;
3. Pip install Pomegranate;
4. Instalar possíveis dependências como numpy e matplotlib;

Agora estamos prontos para programar!

Utilizar é também bastante fácil. Uma vez que a linguagem é um package de python, basta fazer import pomegranate no início de cada script ou na central de comandos se estivermos a usar o interpretador.

Eficiência:

A linguagem é escrita em Cython, que consiste numa mistura entre C e Python, oferecendo desempenhos parecidos com os da linguagem C mas com código que é escrito maioritariamente em Python. Portanto, pomegranate é bastante eficiente.

Numa primeira instância eu tinha escolhido a linguagem Stan, mais concretamente, a versão PyStan que é baseada em Python, mas essa linguagem tinha um grande problema em termos de eficiência para problemas simples. Uma vez que o modelo era compilado em C++, o processo de compilar o código python, encontrar o g++ e compilar esse código era muito lento, demorando, em média, vários minutos para compilar um modelo simples como uma distribuição uniforme. A vantagem de Stan consistia no facto de ser bastante eficiente para modelos complexos uma vez que C++ é uma linguagem bastante eficiente.

Pomegranate tem ainda algumas características que a tornam bastante eficiente:

1. Paralelismo Multi-thread: É uma forma de dividir o trabalho pelo “CPU”, fazendo com que várias threads possam correr ao mesmo tempo;
2. Multiplicação de matrizes aceleradas pelo GPU: Todas as operações que envolvem distribuições gaussianas multivariáveis e todos os modelos que as envolvam são acelerados pela GPU, tornando estes modelos até 4 vezes mais rápidos que os obtidos usando operações lineares algébricas.