📖 **aksnzhy** / **xlearn**

High Performance, Easy-to-use, and Scalable Machine Learning Package

#machine-learning #statistics #data-science #data-analysis

🕐 **645** commits	⑂ **1** branch	🏷 **0** releases	👥 **9** contributors	⚖ Apache-2.0

Branch: **master** ▾   New pull request   Create new file   Upload files   Find file   Clone or download ▾

aksnzhy update doc	Latest commit `2ba1ca6` 6 hours ago

| 📁 demo	update demo	3 days ago
📁 doc	update doc	6 hours ago
📁 gtest	update	4 months ago
📁 img	Fix typo in image	2 days ago
📁 python-package	Merge pull request #28 from purkyston/master	7 hours ago
📁 scripts	update run_example.sh	6 hours ago
📁 src	update ftrl	7 hours ago
📄 .gitignore	update	20 days ago
📄 CMakeLists.txt	update: CMakelist	21 hours ago
📄 LICENSE	Initial commit	6 months ago
📄 README.md	change Release Note to News	3 days ago

📖 **README.md**

![license Apache 2] ![version 0.1.0] ![build passing]

[Installation](#) | [Tutorial](#) | [RoadMap](#) | [News](#)

What is xLearn?

xLearn is a **high performance**, **easy-to-use**, and **scalable** machine learning package, which can be used to solve large-scale classification and regression problems. If you are the user of liblinear, libfm, or libffm, now the xLearn is your another better choice. This project comes from the PKU-Cloud lab: [homepage](#)

## Performance
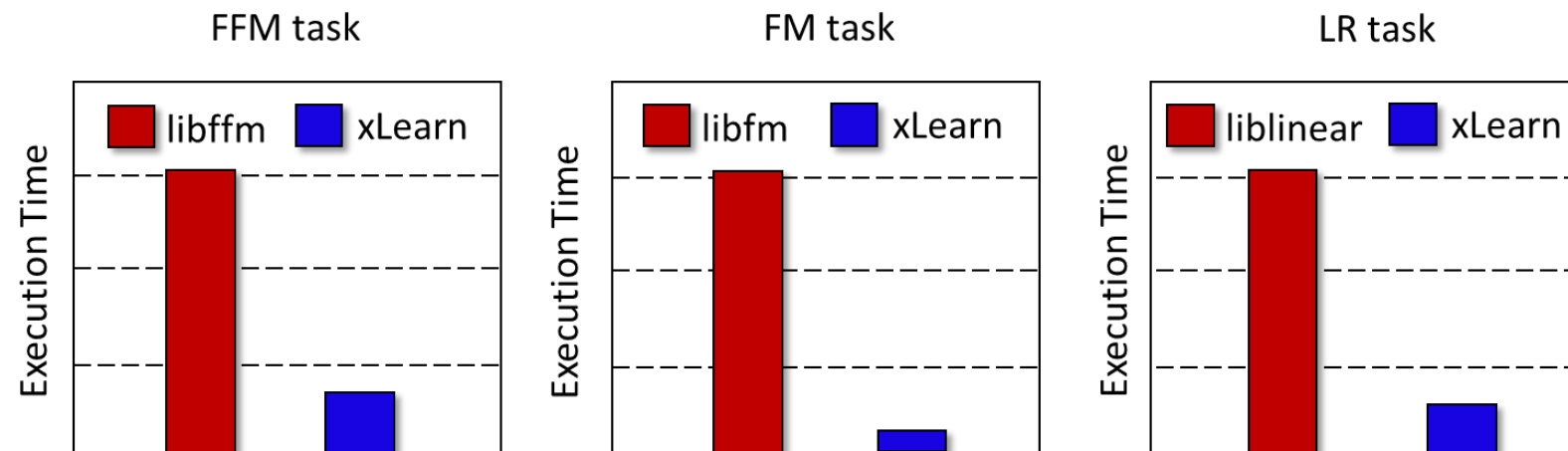
Test on a single MacBook Pro

xLearn is developed by high-performance C++ code with careful design and optimizations. Our system is designed to maximize the CPU and memory utilizations, provide cache-aware computation, and support lock-free learning. By combining these insights, xLearn is 5x - 13x faster compared to the similar systems.

Ease-of-use

```
 1    import xlearn as xl
 2    # Create factorization machine
 3    ffm_model = xl.create_ffm()
 4
 5    # Set training data and validation data
 6    ffm_model.setTrain("./train.txt")
 7    ffm_model.setValidate("./validate.txt")
 8
 9    # Set hyper-parameters
10    param = { 'task':'binary',
11              'lr' : 0.2,
12              'lambda' : 0.002,
13              'metric' : 'auc' }
14
15    # Train model
16    ffm_model.fit(param, "./model.out")
17
18    # Predict
19    ffm_model.setTest("./test.txt")
20    ffm_model.predict("./model.out", "./output.txt")
```

xLearn does not rely on any third-party library, and hence users can just clone the code and compile it by using cmake. Also, xLearn supports very simple python API for users. Apart from this, xLearn supports many useful features that has been widely used in the machine learning competitions like cross-validation, early-stop, etc.

Scalability
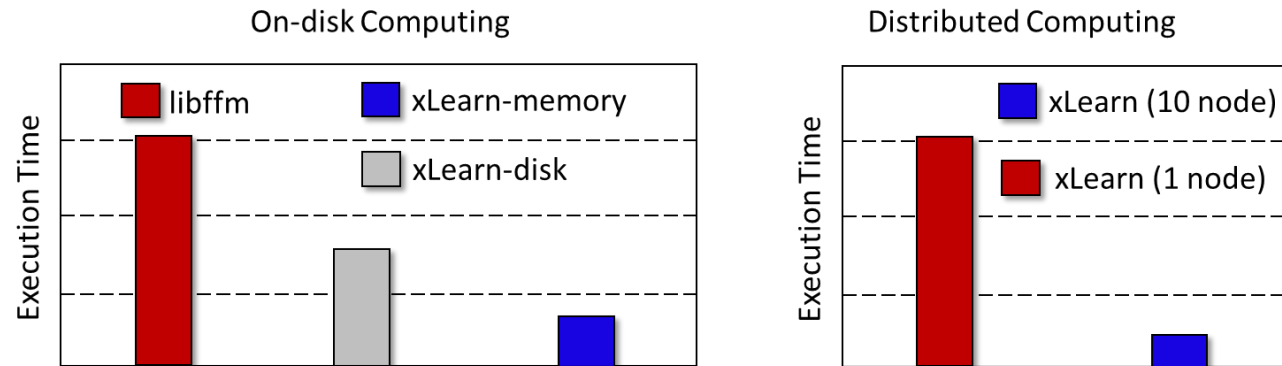
xLearn can be used for solving large-scale machine learning problems. First, xLearn supports out-of-core training, which can handle very large data (TB) by just leveraging the disk of a single machine. Also, xLearn can support distributed training, which scales beyond billions of example across many machines.