📮 komiya-atsushi / xgboost-predictor-java

Pure Java implementation of XGBoost predictor for online prediction tasks.

#java #xgboost #machine-learning

94 commits				44 CC	4 contributors		ুঁা Apache-2.0	
Branch: master ▼	New pull request			Create new file	Upload files	Find file	Clone or download	
komiya-atsus	hi Update documents	6				Latest com	ımit f7e04ca on 25 Jur	
buildSrc/src/main		Introduce xgboost-predictor-spark					9 months ago	
gradle/wrapper		Update Gradle wrapper: 2.14.1				10 months ago		
xgboost-predictor-examples		Remove duplicate test data in xgboost-predictor-example				7 months ago		
xgboost-predictor-spark		Remove duplicate test data in xgboost-predictor-example				7 months ago		
xgboost-predictor-test		Add test data for rank:pairwise				5 months ago		
xgboost-predictor		[skip ci] [Gradle Release Plugin] - new version commit: 'xgboost-pred				5 months ago		
gitignore		Ignore tmp director	·у				5 months ago	
travis.yml		Add .travis.yml				2 years ag		
LICENSE		Add license and readme				2 years ag		
README.md		Update documents					5 months ago	
RELEASENOTES.md		Update documents				5 months ago		
build.gradle		Modify build.gradle	Modify build.gradle				9 months ag	

gradle.properties	Prepare to release 0.2.0	9 months ago
gradlew	Update Gradle wrapper: 2.14.1	10 months ago
gradlew.bat	Add gradle wrapper	2 years ago
settings.gradle	Introduce xgboost-predictor-test	7 months ago

README.md

xgboost-predictor-java

build passing Download 0.2.1

Pure Java implementation of XGBoost predictor for online prediction tasks.

Getting started

Adding to dependencies

If you use Maven:

```
<repositories>
  <repository>
    <id>bintray-komiya-atsushi-maven</id>
    <url>http://dl.bintray.com/komiya-atsushi/maven</url>
  </repository>
</repositories>
</dependencies>
```

```
<dependency>
     <groupId>biz.k11i
     <artifactId>xgboost-predictor</artifactId>
     <version>0.2.1
   </dependency>
 </dependencies>
Or Gradle:
  repositories {
     // Use jcenter instead of mavenCentral
     jcenter()
 dependencies {
     compile group: 'biz.k11i', name: 'xgboost-predictor', version: '0.2.1'
Or sbt:
 resolvers += Resolver.jcenterRepo
 libraryDependencies ++= Seq(
   "biz.k11i" % "xgboost-predictor" % "0.2.1"
```

Using Predictor in Java

```
package biz.k11i.xgboost.demo;
import biz.k11i.xgboost.Predictor;
```

```
import biz.k11i.xgboost.util.FVec;
public class HowToUseXgboostPredictor {
   public static void main(String[] args) throws java.io.IOException {
        // If you want to use faster exp() calculation, uncomment the line below
        // ObjFunction.useFastMathExp(true);
        // Load model and create Predictor
        Predictor predictor = new Predictor(
                new java.io.FileInputStream("/path/to/xgboost-model-file"));
        // Create feature vector from dense representation by array
        double[] denseArray = {0, 0, 32, 0, 0, 16, -8, 0, 0, 0};
        FVec fVecDense = FVec.Transformer.fromArray(
                denseArray,
                true /* treat zero element as N/A */);
        // Create feature vector from sparse representation by map
        FVec fVecSparse = FVec.Transformer.fromMap(
                new java.util.HashMap<Integer, Double>() {{
                    put(2, 32.);
                    put(5, 16.);
                    put(6, -8.);
                }});
        // Predict probability or classification
        double[] prediction = predictor.predict(fVecDense);
        // prediction[0] has
             probability ("binary:logistic")
            class label ("multi:softmax")
        // Predict leaf index of each tree
        int[] leafIndexes = predictor.predictLeaf(fVecDense);
        // leafIndexes[i] has a leaf index of i-th tree
```

```
}
```

Apache Spark integration

See detail xgboost-predictor-spark.

Benchmark

Throughput comparison to xgboost4j 1.1 by xgboost-predictor-benchmark.

Feature	xgboost-predictor	xgboost4j	
Model loading	49017.60 ops/s	39669.36 ops/s	
Single prediction	6016955.46 ops/s	1018.01 ops/s	
Batch prediction	44985.71 ops/s	5.04 ops/s	
Leaf prediction	11115853.34 ops/s	1076.54 ops/s	

Xgboost-predictor-java is about 6,000 to 10,000 times faster than xgboost4j on prediction tasks.

Supported models, objective functions and API

- Models
 - o "gblinear"
 - o "gbtree"
- · Objective functions

- o "binary:logistic"
- "binary:logitraw"
- "multi:softmax"
- o "multi:softprob"
- o "reg:linear"
- o "rank:pairwise"
- API
 - Predicts probability or classification
 - Predictor#predict(FVec)
 - Outputs margin
 - Predictor#predict(FVec, true /* output margin */)
 - Predicts leaf index
 - Predictor#predictLeaf(FVec)