朴素贝叶斯是一种简单的多类分类算法，假设每对特征之间具有独立性。朴素贝叶斯可以非常有效地训练。在训练数据的单次传递中，它计算给定标签的每个特征的条件概率分布，然后应用贝叶斯定理来计算给定观测值的标签的条件概率分布并将其用于预测。

spark.mllib支持多项式朴素贝叶斯和伯努利朴素贝叶斯。这些模型通常用于文档分类。在该上下文中，每个观察是一个文档，每个特征代表一个术语，其值是术语的频率（在多项式朴素贝叶斯中），0或1表示该术语是否在文档中找到（在伯努利朴素贝叶斯中）。特征值必须是非负的。使用可选参数“multinomial”或“bernoulli”选择模型类型，默认为“multinomial”。 通过设置参数λ（默认为1.0）可以使用平滑。对于文档分类，输入特征向量通常是稀疏的，并且应该提供稀疏向量作为输入以利用稀疏性。由于训练数据仅使用一次，因此不必对其进行缓存。

NaiveBayes类实现了多项式朴素贝叶斯。它采用LabeledPoint的RDD和可选的平滑参数lambda作为输入，可选的模型类型参数（默认为“multinomial”），并输出NaiveBayesModel，可用于评估和预测。

|  |
| --- |
| 代码示例： |
| **package** com.cb.spark.mllib;  **import** org.apache.spark.SparkConf;  **import** org.apache.spark.api.java.JavaPairRDD;  **import** org.apache.spark.api.java.JavaRDD;  **import** org.apache.spark.api.java.JavaSparkContext;  **import** org.apache.spark.mllib.classification.NaiveBayes;  **import** org.apache.spark.mllib.classification.NaiveBayesModel;  **import** org.apache.spark.mllib.regression.LabeledPoint;  **import** org.apache.spark.mllib.util.MLUtils;  **import** scala.Tuple2;  **public** **class** JavaNaiveBayesExample {  **public** **static** **void** main(String[] args) {  SparkConf conf = **new** SparkConf().setAppName("JavaNaiveBayesExample").setMaster("local");  JavaSparkContext jsc = **new** JavaSparkContext(conf);  String path = "F:\\Learning\\java\\project\\LearningSpark\\src\\main\\resources\\sample\_libsvm\_data.txt";  JavaRDD<LabeledPoint> inputData = MLUtils.*loadLibSVMFile*(jsc.sc(), path).toJavaRDD();  JavaRDD<LabeledPoint>[] tmp = inputData.randomSplit(**new** **double**[] { 0.6, 0.4 });  JavaRDD<LabeledPoint> training = tmp[0];  JavaRDD<LabeledPoint> test = tmp[1];  NaiveBayesModel model = NaiveBayes.*train*(training.rdd(), 1.0); // 1.0为平滑参数  JavaPairRDD<Double, Double> predictionAndLabel = test  .mapToPair(p -> **new** Tuple2<>(model.predict(p.features()), p.label()));  **double** accuracy = predictionAndLabel.filter(pl -> pl.\_1.equals(pl.\_2())).count() / (**double**) test.count();  System.***out***.println("模型测试精度为:" + accuracy);  }  } |