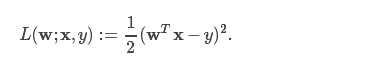
# 1. 线性最小二乘，Lasso回归和岭回归

线性最小二乘法是回归问题的最常见公式。损失函数如下：



通过使用不同类型的正则化来导出各种相关的回归方法：普通最小二乘或线性最小二乘不使用正则化; 岭回归使用L2正则化; Lasso使用L1正则化。对于所有这些模型，平均损失或训练误差，



被称为均方误差。

MLlib的所有方法都使用Java友好类型，因此您可以像在Scala中一样导入和调用它们。唯一需要注意的是，这些方法使用Scala RDD对象，而Spark Java API使用单独的JavaRDD类。您可以通过在JavaRDD对象上调用.rdd（）将Java RDD转换为Scala。下面提供了Scala代码段的相应Java示例：

|  |
| --- |
| SparkConf conf = **new** SparkConf().setAppName("JavaLinearRegressionWithSGDExample").setMaster("local");  JavaSparkContext sc = **new** JavaSparkContext(conf);  String path = "F:\\Learning\\java\\project\\LearningSpark\\src\\main\\resources\\lpsa.data";  JavaRDD<String> data = sc.textFile(path);  JavaRDD<LabeledPoint> parsedData = data.map(line -> {  String[] parts = line.split(",");  String[] features = parts[1].split(" ");  **double**[] v = **new** **double**[features.length];  **for** (**int** i = 0; i < features.length - 1; i++) {  v[i] = Double.*parseDouble*(features[i]);  }  **return** **new** LabeledPoint(Double.*parseDouble*(parts[0]), Vectors.*dense*(v));  });  parsedData.cache();  // 模型构建和训练  **int** numIterations = 100; // 迭代次数  **double** stepSize = 0.00000001; // 学习率  LinearRegressionModel model = LinearRegressionWithSGD.*train*(JavaRDD.*toRDD*(parsedData), numIterations, stepSize);  JavaPairRDD<Double, Double> valueAndPreds = parsedData  .mapToPair(point -> **new** Tuple2<>(model.predict(point.features()), point.label()));  **double** MSE = valueAndPreds.mapToDouble(pair -> {  **double** diff = pair.\_1() - pair.\_2();  **return** diff \* diff;  }).mean();  System.***out***.println("训练数据的均方误差为:" + MSE); |

# 2. Streaming linear regression

当数据以流式方式到达时，在线拟合回归模型，在新数据到达时更新模型的参数是有用的。spark.mllib目前支持使用普通最小二乘法进行流线性回归。拟合类似于离线执行的拟合，除了在每批数据上进行拟合，因此模型不断更新以反映来自流的数据。以下示例演示如何从两个不同的文本文件输入流加载训练和测试数据，将流解析为标记点，将线性回归模型在线拟合到第一个流，并对第二个流进行预测。

首先，我们导入必要的类来解析输入数据并创建模型。然后我们为训练和测试数据制作输入流。我们假设已经创建了StreamingContext ssc。对于此示例，我们在训练和测试流中使用标记点，但实际上您可能希望使用未标记的矢量来测试数据。我们通过将权重初始化为零来创建我们的模型，并注册流以进行训练和测试然后开始工作。输出预测与真实标签一起使我们可以轻松查看结果。

最后，我们可以将包含数据的文本文件保存到训练或测试文件夹中。每一行应该是格式为（y，[x1，x2，x3]）的数据点，其中y是标签，x1，x2，x3是特征。只要文本文件放在args（0）中，模型就会更新。只要文本文件放在args（1）中，您就会看到预测。当您将更多数据提供给训练文件夹时，预测会变得更好！

|  |
| --- |
| import org.apache.spark.mllib.linalg.Vectors  import org.apache.spark.mllib.regression.LabeledPoint  import org.apache.spark.mllib.regression.StreamingLinearRegressionWithSGD  val trainingData = ssc.textFileStream(args(0)).map(LabeledPoint.parse).cache()  val testData = ssc.textFileStream(args(1)).map(LabeledPoint.parse)  val numFeatures = 3  val model = new StreamingLinearRegressionWithSGD()  .setInitialWeights(Vectors.zeros(numFeatures))  model.trainOn(trainingData)  model.predictOnValues(testData.map(lp => (lp.label, lp.features))).print()  ssc.start()  ssc.awaitTermination() |