6-1-cn

1

大家好，我是来自北京理工大学计算机学院

数据科学与知识工程研究所的车海莺

在这节课我们讨论 Spark MLlib 架构。

2

数据处理系统提供大数据计算处理能力和应用开发平台。从计算架构的角度，将数据处理系统分为数据算法层、计算模型层、计算平台层、计算引擎层等。

与大数据相关的计算算法包括机器学习算法和数据挖掘算法。计算模型是指不同类型的大数据在不同场景下的处理方式，包括批处理、流计算、结构化数据

的大规模并发处理(MPP)模型、内存计算模型和数据流图模型。

计算平台和引擎**提供各种开发套件和操作环境**，我们选取 Spark Mllib 和 TensorFlow 为例，本节我们来讨论 Spark Mllib

3

Apache Spark 的机器学习库(MLlib)旨在简化、可伸缩性和易于与其他工具集成。

通过 Spark 的可伸缩性、语言兼容性和速度，数据科学家可以专注于他们的数据问题和模型，而不是解决围绕分布式数据的复杂问题(如基础设施、配置等)。

Spark MLlib 与 Spark SQL、Spark Streaming 和 DataFrames API 等其他 Spark 组件无缝集成。

机器学习可以应用于各种各样的数据类型，如向量、文本、图像和结构化数据。

该 API 采用来自 Spark SQL 的数据框架，以支持多种数据类型。

该库可以作为 Spark 应用程序的一部分在 Java、Scala 和 Python 中使用，因此您可以将它包含在完整的工作流中。

MLlib 允许预处理、修改咀嚼、训练模型和对数据进行大规模预测。

您甚至可以使用 MLlib 中训练过的模型在结构化流数据中进行预测。

4

使用 spark MLlib 的大数据分析包括描述性分析和基于描述性分析数据和预测分析数据的预测分析。

在描述性分析的范围内，它支持统计描述分析和聚类。

在预测分析的范围内，它支持特征建模，包括特征提取器如 TF-IDF，特征转换器如 Vector Slicer 等和特征选择器如卡方选择器。它还支持二元分类、多类分类和回归等预测算法。

5 为什么使用 Spark Mllib

MLlib 是 Apache Spark 的可扩展机器学习库。

1）的易用性

可用于 Java、Scala、Python 和 R。

MLlib 适合 Spark 的 api，并与 Python 中的 NumPy(从 Spark 0.9 开始)和 R 库(从 Spark 1.5 开始)进行互操作。

您可以使用任何 Hadoop 数据源(例如 HDFS、HBase 或本地文件)，使其易于插入 Hadoop 工作流。

2）性能

高质量算法，速度比 MapReduce 快 100 倍。

Spark 擅长迭代计算，使 MLlib 能够快速运行。

同时，我们关心算法性能:MLlib 包含利用迭代的高质量算法，可以产生比 MapReduce 更好的结果。

3）随处运行

Spark 运行在 Hadoop、Apache Mesos、Kubernetes 上，独立运行，或者在云中针对不同的数据源运行。

您可以在 EC2、Hadoop YARN、Mesos 或 Kubernetes 上使用其独立集群模式运行 Spark。可访问 HDFS、Apache Cassandra、Apache HBase、Apache Hive 等数百个数据源中的数据。

为了支持使用 Spark 的 Python, Apache Spark 社区发布了一个工具 PySpark。使用 PySpark，可以使用 Python 编程语言中的 rdd。

6
构建在 Spark 之上的 MLlib 是一个可扩展的机器学习库，由 4 个主要组件组成:算法、特性、管道和实用程序。
常见的学习算法包括分类、回归、聚类、协同过滤、降维和底层优化原语。
以及实用课程，包括线性代数、统计学等。
特征化包括特征的提取和特征的变换。
ML pipeline 提供了一组构建在数据框架之上的统一的高级 api，帮助用户创建和调优实用的机器学习管道。

完成 ML 任务的机制是构建管道来完成所有需要的步骤。如建立模型、训练模型、评估模型、参数调优和模型持久化。
7
Spark Mllib 支持的支持的机器学习算法包括::

- Classification: logistic regression, naive Bayes,...
- Regression: generalized linear regression, survival regression,...
- Decision trees, random forests, and gradient-boosted trees
- Recommendation: alternating least squares (ALS)
- Clustering: K-means, Gaussian mixtures (GMMs),...
- Topic modeling: latent Dirichlet allocation (LDA)
- Frequent item sets, association rules, and sequential pattern mining

8 **Spark MLlib 工作流实用程序**
机器学习的实用程序包括:

- **特征转换 Feature transformations**:标准化、规范化、哈希……
- **ML Pipeline 机器学习管道构建**
- **Model evaluation and hyper-parameter tuning 模型评估和超餐调优**
- **ML 机器学习模型和 pipeline 的持久化**: 存储和下载训练好的模型和制作好的管道 pipeline

其他实用程序:

- **Distributed linear algebra 分布式线性代数方法**: SVD, PCA,...
- **Statistics 统计**: 汇总统计，假设检验，…

9 机器学习 Pipeline
左边是正常的机器学习流水线 Pipeline，
1）加载/清理数据，2）特征提取，3）模型训练，4）模型评估和参数调优，然后重复工作流过程。
右边是 Spark Mllib Pipeline 概念，
从 1）加载/清洁数据，2）转换器，对应特征工程，3）Estimator，对应模型训练，4）Evaluator，负责模型评估。下面逐一解释
10
Transformer 是一种抽象，包括特征转换器和学习模型。
将数据转换为可使用的格式，
取输入列，将其转换为输出列。
从技术上讲，Transformer 实现了 transform()方法，它将一个 DataFrame 转换为另一个 DataFrame，通常是通过附加一个或多个列。
例如:
特征转换器可以取一个 DataFrame，读取一个列(例如，文本)，将它映射到一个新的列(例如，特征向量)并输出一个新的 DataFrame，其中附加了映射的列。
学习模型可以取一个 DataFrame，读取包含特征向量的列，预测每个特征向量的标签，然后输出一个新的 DataFrame，其中预测的标签作为列追加。

例如:1 规范化数据，2 标记化(这意味着将句子分成单词)和 3 将分类值转换为数字。
如图所示，将数据帧 1 转换为数据帧 2。

11 Estimator 预测器

预测器抽象了学习算法或任何训练数据的算法的概念。
从技术上讲，Estimator 实现一个 fit()方法，该方法接受一个 DataFrame 并生成一个 Model，是一个 Transformer。
根据数据进行训练(拟合)的学习算法
返回一个 Transformer 类型的模型
例如，像 LogisticRegression 这样的学习算法是一个估计器，调用 fit()训练一个 LogisticRegressionModel，它是一个 Model，因此是一个 Transformer。

12 Evaluator 评估器

基于某些指标用来评估模型的性能设计评估器，如 ROC, RMSE。
通过比较模型性能评估器可以帮助自动化模型调优过程
选择产生预测的最佳模型。
这里的例子是 BinaryClassificationEvaluator 与 CrossValidator。
输入是数据，输出是从所有选项中选择的最佳模型。

13 Pipeline

在机器学习中，通常会运行一系列算法来处理和学习数据。
例如，一个简单的文本文档处理工作流可能包括以下几个阶段:
将每个文档的文本拆分为文字。
将每个文档的单词转换为一个数字特征向量。
学习使用特征向量和标签的预测模型。
MLlib 将这样一个工作流表示为管道，它由一系列以特定顺序运行的 PipelineStages (transformer 和 Estimators)组成。
该管道利用了变压器和估计器的统一 API。它可以被坚持。
这些阶段按顺序运行，输入的 DataFrame 在经过每个阶段时进行转换。
对于 Transformer 阶段，在 DataFrame 上调用 transform()方法。
对于 Estimator 阶段，fit()方法被调用以生成 Transformer(它成为 PipelineModel 或拟合的 Pipeline 的一部分)，而 Transformer 的 transform()方法在 DataFrame 上被调用。
我们为简单的文本文档工作流演示了这一点。
中间的图，即顶部行表示一个包含三个阶段的 Pipeline。
前两个(Tokenizer 和 HashingTF)是 transformer(蓝色)，第三个(LogisticRegression)是 Estimator(红色)。
底部一行表示流经管道的数据，其中圆柱表示 DataFrames。
在原始的 DataFrame 上调用 pipe .fit()方法，该 DataFrame 具有原始文本文档和标签。
transform()方法将原始文本文档拆分为单词，向 DataFrame 添加一个包含单词的新列。
transform()方法将单词列转换为特征向量，并将包含这些向量的新列添加到 DataFrame 中。
现在，由于 LogisticRegression 是一个估计器，管道首先调用 LogisticRegression.fit()来生成一个 LogisticRegressionModel。
如果管道有更多的估计器，它将调用 DataFrame 上的 LogisticRegressionModel 的 transform()方法，然后将 DataFrame 传递到下一个阶段。
管道是一个估计器。
因此，在 Pipeline 的 fit()方法运行之后，它会生成一个 PipelineModel，这是一个 Transformer(这意味着模型是 Transformer)。
这个 PipelineModel 在测试时使用;下面的图说明了这种用法。
PipelineModel 具有与原始管道相同的阶段数，但是原始管道中的所有估计器都变成了 transformer。
当在测试数据集上调用 PipelineModel 的 transform()方法时，数据将按顺序通过拟合的管道。

每个阶段的 transform()方法更新数据集并将其传递给下一个阶段。
管道和 pipelinmodels 帮助确保训练和测试数据经过相同的特征处理步骤。

14 参数 Parameters

Mllib 预测器和变形器使用统一的 API 来指定参数。
Param 是一个具有自包含文档的命名参数。
ParamMap 是一组(参数，值)对。
向算法传递参数主要有两种方式:
设置实例参数。例如，如果 lr 是 LogisticRegression 的一个实例，可以调用 lr. setmaxiter(10)(最大迭代)，使 lr.fit()最多使用 10 次迭代。这个 API 类似于 spark 中使用的 API。mllib 包。
传递一个 ParamMap 给 fit()或 transform()。ParamMap 中的任何参数都将覆盖之前通过 setter 方法指定的参数。
参数属于估计器和转换器的特定实例。
例如，如果我们有两个 LogisticRegression 实例 lr1 和 lr2，那么我们可以用指定的 maxIter 参数构建一个 ParamMap: maxIter -> 10, lr2.maxIter - > 20)。如果在一个 Pipeline 中有两个带有 maxIter 参数的算法，这将非常有用。

15 自动化模型调优过程

在 MLlib 中构建管道之后，它可以使模型调优过程自动化。
ML 中的一个非常重要的任务是模型选择，或者说，使用数据为给定的任务找到最佳的模型或参数。
这被称为参数调优。可以在单个估计器(如 LogisticRegression)上执行调优
整个流程(可以包括多个算法、特征描述和其他步骤)。
我们需要为基于网格搜索的模型选择建立一个参数网格，
为了构建参数网格，我们可以使用 ParamGridBuilder 工具类。
ParamGridBuilder()允许为单个参数指定不同的值，
然后比较整个参数集选择最佳方案，从而确定最佳模型。
CrossValidator 将数据集分成几个折叠，这些折叠可用于独立的训练和测试集。
例如:当 k=5 折叠时，CrossValidator 将生成 5 个(训练，测试)对，每个对使用 4/5 数据作为训练集，1/5 数据作为测试集。
为了评估一个特定的 ParamMap，使用 Estimator 在 5 个不同的数据对上拟合 5 个模型，而 CrossValidator 将计算 5 个评估指标的平均值。
选择最佳 ParamMap 后，
CrossValidator 将最终使用相应的 Estimator 和最佳的 ParamMap 来改装整个数据集。

16 模型持久化

ML 持久性意味着保存和加载管道
通常情况下，将模型或管道保存到磁盘以备以后使用是值得的。
数据科学家创建模型或管道，数据工程师可以大规模部署模型并监视其应用。
在 Spark 1.6 中，一个模型导入/导出功能被添加到 Pipeline API 中。
从 Spark 2.3 开始，Spark 中基于数据帧的 API。毫升和 pyspark。MI 的覆盖范围很广。
ML 持久性可以跨 Scala、Java 和 Python 工作。
然而，R 目前使用的是修改后的格式，所以保存在 R 中的模型只能加载回 R 中;这应该在未来被修复，并在 SPARK-15572 中被跟踪。

17 动手实验

考虑到 Spark 的 ML Lib 适用于对大型数据集进行相对简单的 ML 操作。
ML Lib 对于小型数据集的计算效率不高，您最好使用 scikit-learn 来处理中小型数据集(兆字节，最多可达几 gb)。
我们在 sklearn 上进行了实践，其中包括算法和高级工具。
Scikit-learn (Sklearn)是 Python 中用于机器学习的最有用、最健壮的库。

它为机器学习和统计建模提供了一系列有效的工具，包括通过 Python 中的一致性接口进行分类、回归、聚类和降维。

18 本节我们学习了 spark MLlib，今天的内容就到这里，谢谢大家