

Spark推荐系统

Spark Recommendation System

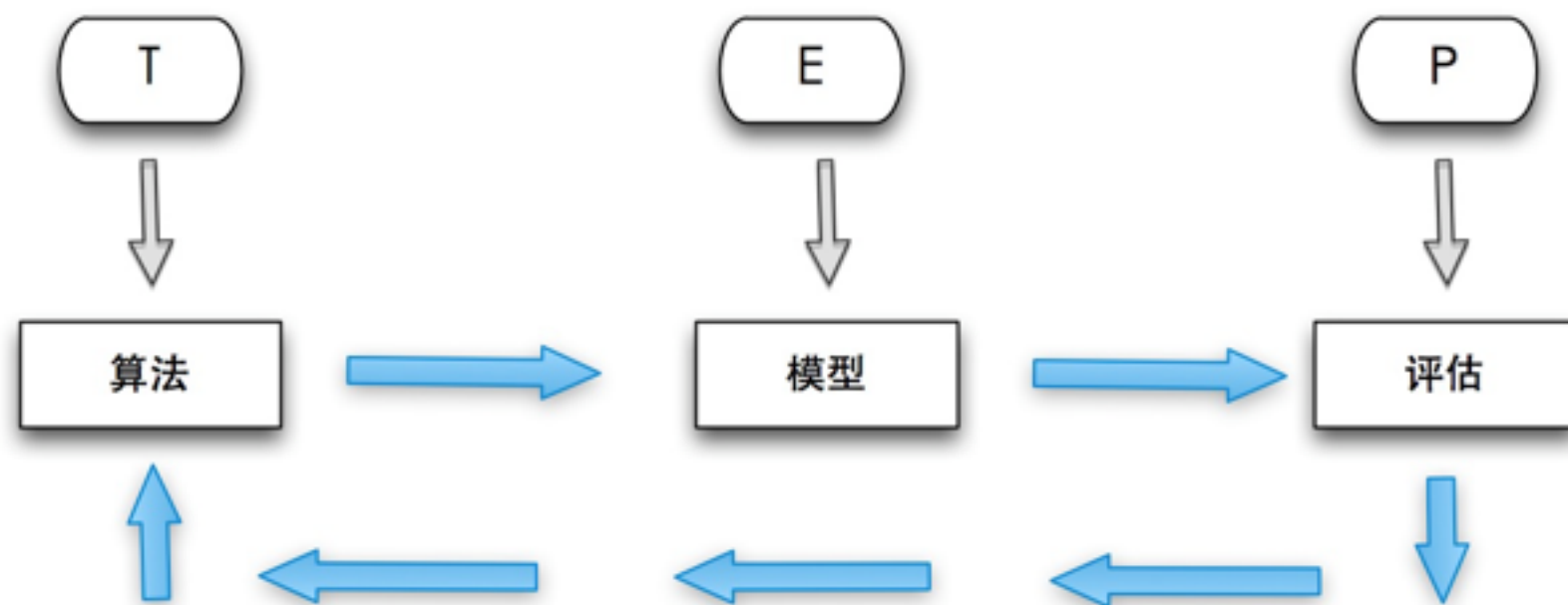
Spark MLlib

Spark MLlib简介

- 什么是机器学习
- 基于大数据的机器学习
- Spark 机器学习库MLLib

什么是机器学习

- 机器学习可以看做是一门人工智能的科学，该领域的主要研究对象是人工智能。机器学习利用数据或以往的经验，以此优化计算机程序的性能标准。
- 机器学习强调三个关键词：算法、经验、性能



基于大数据的机器学习

- 传统的机器学习算法，由于技术和单机存储的限制，只能在少量数据上使用，依赖于数据抽样
- 大数据技术的出现，可以支持在全量数据上进行机器学习
- 机器学习算法涉及大量迭代计算
- 基于磁盘的MapReduce不适合进行大量迭代计算
- 基于内存的Spark比较适合进行大量迭代计算

Spark的机器学习库

- Spark提供了一个基于海量数据的机器学习库，它提供了常用机器学习算法的分布式实现
- 开发者只需要有 Spark 基础并且了解机器学习算法的原理，以及方法相关参数的含义，就可以轻松的通过调用相应的 API 来实现基于海量数据的机器学习过程
- Spark-Shell也是一个关键。算法工程师可以边写代码边运行，边看结果

Spark的机器学习库

- MLlib是Spark的机器学习（Machine Learning）库，旨在简化机器学习的工程实践工作
- MLlib由一些通用的学习算法和工具组成，包括分类、回归、聚类、协同过滤、降维等，具体如下：
 - 算法工具：常用的学习算法，如分类、回归、聚类和协同过滤；
 - 特征化工具：特征提取、转化、降维和选择工具；
 - 工作流(Pipeline)：用于构建、评估和调整机器学习工作流的工具；
 - 持久性：保存和加载算法、模型和管道；
 - 实用工具：线性代数、统计、数据处理等工具。

Spark的机器学习库

- Spark 机器学习库从1.2 版本以后被分为两个包：
- spark.mllib 包含基于RDD的原始算法API。Spark MLlib 历史比较长，在1.0 以前的版本即已经包含了，提供的算法实现都是基于原始的 RDD
- spark.ml 则提供了基于DataFrames 高层次的API，可以用来构建机器学习工作流（PipeLine）。ML Pipeline 弥补了原始 MLlib 库的不足，向用户提供了一个基于 DataFrame 的机器学习工作流式 API 套件

Spark的机器学习库

- MLlib目前支持4种常见的机器学习问题: 分类、回归、聚类
和协同过滤

	离散数据	连续数据
监督学习	Classification、 LogisticRegression(with Elastic-Net)、 SVM、DecisionTree、 RandomForest、GBT、NaiveBayes、 MultilayerPerceptron、OneVsRest	Regression、 LinearRegression(with Elastic- Net)、DecisionTree、 RandomFores、GBT、 AFTSurvivalRegression、 IsotonicRegression
无监督学习	Clustering、KMeans、 GaussianMixture、LDA、 PowerIterationClustering、 BisectingKMeans	Dimensionality Reduction, matrix factorization、PCA、SVD、ALS、 WLS

Spark 推荐系统

Spark推荐系统

- 使用Spark训练一个推荐模型，对电影进行推荐
- 数据集： MovieLens
- <https://grouplens.org/datasets/movielens/>

Spark推荐系统

movielens

Non-commercial, personalized movie recommendations.

[sign up now](#)

or [sign in](#)

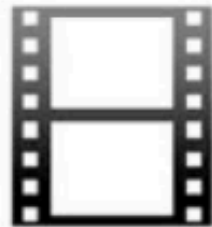
recommendations

MovieLens helps you find movies you will like. Rate movies to build a custom taste profile, then MovieLens recommends other movies for you to watch.

The screenshot displays the 'top picks' section of the MovieLens website. At the top, the text 'top picks' is followed by a 'see more' button. Below this, a subtitle reads 'based on your ratings, MovieLens recommends these movies'. A row of seven movie cards is shown, each with a title, year, rating, and duration. The movies are: 'Band of Brothers' (2001, R, 705 min), 'Casablanca' (1942, PG, 102 min), 'One Flew Over the Cuckoo's Nest' (1975, R, 133 min), 'The Lives of Others' (2006, R, 137 min), 'Sunset Boulevard' (1950, NR, 110 min), 'The Third Man' (1949, NR, 104 min), and 'Patton' (1957). Each card features a movie poster and a star rating bar at the bottom.

Movie Title	Year	Rating	Duration
Band of Brothers	2001	R	705 min
Casablanca	1942	PG	102 min
One Flew Over the Cuckoo's Nest	1975	R	133 min
The Lives of Others	2006	R	137 min
Sunset Boulevard	1950	NR	110 min
The Third Man	1949	NR	104 min
Patton	1957		

Collaborative Filtering



★	★★★★	?
★	★★★	★★
★★★★	?	★
★	?	★★
?	★★★	★★
★★★★	★★	?

last.fm



LinkedIn

amazon.com

Spark Matrix Factorization

- Spark推荐系统只支持基于矩阵分解的实现
- 这类模型在协同过滤中的表现十分出色
- 在Netflix Prize等知名的比赛中表现也很突出

Spark Matrix Factorization

- 显式矩阵分解
 - 当要处理的那些数据是由用户提供的自身的偏好数据，这类数据称为显式偏好数据，例如

	无间道	蜘蛛侠	惊奇队长
A同学	3	3	
B同学		2	4
C同学		5	

Spark Matrix Factorization

- 在实际中刚才的矩阵往往非常稀疏
- 对于这个矩阵建模，可以采用矩阵分解的方式。
- 具体的做法就是找出两个低纬度的矩阵，使得他们的乘积是原始的矩阵

Spark Matrix Factorization

	<i>item 1</i>	<i>item 2</i>	<i>item 3</i>	...	<i>item n</i>
<i>user 1</i>					
<i>user 2</i>					
<i>user 3</i>					
<i>user 4</i>					
<i>user 5</i>					
<i>user 6</i>					
<i>user 7</i>					
<i>user 8</i>					
...					
<i>user n</i>					

R

\approx

	<i>feature 1</i>	<i>feature 2</i>
<i>user 1</i>		
<i>user 2</i>		
<i>user 3</i>		
<i>user 4</i>		
<i>user 5</i>		
<i>user 6</i>		
<i>user 7</i>		
<i>user 8</i>		
...		
<i>user n</i>		

U

\times

	<i>item 1</i>	<i>item 2</i>	<i>item 3</i>	...	<i>item n</i>
<i>feature 1</i>					
<i>feature 2</i>					

V

Spark Matrix Factorization

- 目标函数为

$$J(x_*, y_*) = \sum (r_{ui} - x_u y_i)^2 + \lambda (\sum_u ||x_u||^2 + \sum_i ||y_i||^2)$$

- r 为原矩阵中的元素
- x 为 U 矩阵中的元素
- y 为 Y 矩阵中的元素

Alternating Least Squares, ALS

- ALS 交替最小二乘法是一种求矩阵分解问题的最优化方法。
- 在每一次迭代中先固定 U 或者 V ，来更新另一个矩阵的参数。然后固定更新过的矩阵，再更新另一个
- 直到收敛为止