【Spark Summit East 2017】使用Spark进行时间序列分析 免费注册 (//account.aliyun.com/register/register.htm?oauth_callback=https%3A%2F%2Fyq.aliyun.com%2Farti
(//www.aliyun.com/)

控制台 (//home.console.aliyun.com/) 文档 (//help.aliyun.com/) 备案 (//beian.aliyun.com/) 邮箱 (//qiye.aliyun.com/)
找社区文章、答案、技术大拿 (/notifications) 登录社区 (https://account.aliyun.com/login/login.htm?from_type=yqclub&oauth_callback=https%3A%2F%2Fyq.aliyun.com%2Farticles%2F70292%

登录 (//account.aliyun.com/login/login.htm?qrCodeFirst=false&oauth_callback=https%3A%2F%2Fyq.aliyun.com%2Farticles%2F70292)

(/)

人脸识别

云栖社区 > 博客列表 (/articles) > 正文

云头条 (/cloud) 博客 (/articles) 问答 (/ask) 聚能聊 (/roundtable) 直播 (/webinar) 论坛 (https://bbs.aliyun.com) 云栖大会 (https://yunqi.aliyun.com) 公众号 (/t

订阅 (/publication) 云大学 (https://edu.aliyun.com) 更多


(https://yq.aliyun.com/promotion/543)

# 【Spark Summit East 2017】使用Spark进行时间序列分析

小猫吃鱼569 (/users/1065172560996322) 2017-02-18 15:44:22 浏览3184 评论0

云栖社区 (/tags/type_blog-tagid_1/) python (/tags/type_blog-tagid_14/) 大数据 (/tags/type_blog-tagid_24/) java (/tags/type_blog-tagid_41/) HTTPS (/tags/type_blog-tagid_456/) spark (/tags/type_blog-tagid_1229/) scala (/tags/type_blog-tagid_1250/) aliyun (/tags/type_blog-tagid_1251/) 大数据分析 (/tags/type_blog-tagid_8668/) MaxCompute (/tags/type_blog-tagid_10127/)

**摘要：** 本讲义出自Simon Ouellette在Spark Summit East 2017上的演讲，主要介绍了在Spark上与时间序列数据进行交互的Scala / Java / Python库——spark-timeseries，演讲中分享了spark-timeseries的总体设计，目前实现的功能，并将提供一些用法示例。

更多精彩内容参见**云栖社区大数据频道**https://yq.aliyun.com/big-data (https://yq.aliyun.com/big-data)；此外，通过**Maxcompute及其配套产品**，低廉的大数据分析仅需几步，详情访问https://www.aliyun.com/product/odps (https://www.aliyun.com/product/odps)。

本讲义出自**Simon Ouellette**在Spark Summit East 2017上的演讲，主要介绍了在Spark上与时间序列数据进行交互的Scala / Java / Python库——spark-timeseries，演讲中分享了spark-timeseries的总体设计，目前实现的功能，并将提供一些用法示例。因为项目还处于早期阶段，演讲也介绍了spark-timeseries当前的缺点和未来spark-timeseries项目的发展路线图。


达人介绍

小猫吃鱼569 (/users/1065172
文章 215篇 | 关注
关注

文中提到的云产品

E-MapReduce (/go/1/22?postion=2)
构建于阿里云 ECS 弹性虚拟机之上，利用开源系统，包括 Hadoop、Spark... 查看详情 (/go/1

分析型数据库 (/go/1/24?postion=2)
阿里巴巴自主研发的海量数据实时高并发在线服务，使得您可以在毫秒级针对千亿级数据进行

博主其他文章

更多> (/users/106517256

nit精选分享PDF合
pache Hadoop
)
云上基于Hadoop
构建成功的数据湖
乐高一样搭建
(/articles/71243)
DFS演化成为广义
che Spark &
/71240)
che Storm中的资源
mnar Era：利用
(/articles/71236)
时之内使用Apache

相关话题 更多>

# 【Spark Summit East 2017】使用Spark进行时间序列分析

阿里女神节，20+技术女神盘点，你最想与谁
(/roundtable/65763)
你的工作中有哪些神器工具呢？(/roundtabl
一起来分享你的开年采购季购买计划和攻略!
(/roundtable/65660)
现在是否要开始警惕人工智能了？(/roundta
2018全球十大突破性技术，你知道吗？
(/roundtable/65401)

(https://promotion.aliyun.com/ntms/act/season

## What is spark-timeseries?

https://github.com/sryza/spark-timeseries

- Open source time series library for Apache Spark 2.0
- Sandy Ryza
  - Advanced Analytics with Spark: Patterns for Learning from Data at Scale
  - Senior Data Scientist at Clover Health
- Started in February 2015

## Who am I?

http://faimdata.com

- Chief Data Science Officer at Faimdata
- Contributor to spark-timeseries since September 2015
- Participated in early design discussions (March 2015)
- Been an active user for ~2 years

**Design Question #1**: How do we structure multivariate time series?

Columnar or Row-based?

**Columnar representation**

```
TimeSeriesRDD(
    DateTimeIndex,
    RDD[Vector]
)
```

| DateTime Index | Vector for Series 1 | Vector for Series 2 |
|---|---|---|
| 2:30:01 | 4.56 | 78.93 |
| 2:30:02 | 4.57 | 79.92 |
| 2:30:03 | 4.87 | 79.91 |
| 2:30:04 | 4.48 | 78.99 |

**Row-based representation**

```
RDD[(ZonedDateTime, Vector)]
```

| Vectors | Date/ Time | Series 1 | Series 2 |
|---|---|---|---|
| Vector 1 | 2:30:01 | 4.56 | 78.93 |
| Vector 2 | 2:30:02 | 4.57 | 79.92 |
| Vector 3 | 2:30:03 | 4.87 | 79.91 |
| Vector 4 | 2:30:04 | 4.48 | 78.99 |

【Spark Summit East 2017】使用Spark进行时间序列分析

## Columnar vs Row-based

**More efficient in columnar representation:**

- Lagging
- Differencing
- Rolling operations
- Feature generation
- Feature selection
- Feature transformation

**More efficient in row-based representation:**

- Regression
- Clustering
- Classification
- Etc.

## Example: lagging operation

**Row-based representation**

- Time complexity: O(N) (assumes pre-sorted RDD)
- For each row, we need to get values from previous **k** rows

**Columnar representation**

- Time complexity: O(K)
- For each column to lag, we truncate most recent **k** values, and truncate the DateTimeIndex's oldest **k** values.

## Example: regression

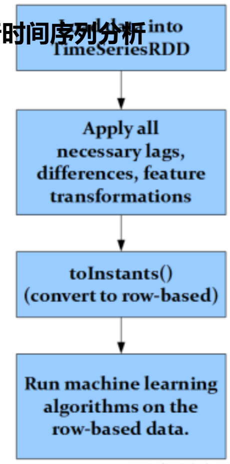
- We're estimating: $y_t = \alpha + \sum_I \sum_J \beta_{ij} x_{i(t-j)}$
- The lagged values are typically part of each row, because they are pre-generated as new features.
- **Stochastic Gradient Descent**: we iterate on examples and estimate error gradient to adjust weights, which means that we care about rows, not columns.
- To avoid shuffling, the partitioning must be done such that all elements of a row are together in the same partition (so the gradient can be computed locally).

## Current solution

- Core representation is columnar.
- Utility functions to go to/from row-based.
- **Reasoning**: spark-timeseries operations are mostly time-related, i.e. columnar. Row-based operations are about relationships between the variables (ML/statistical), thus external to spark-timeseries.

# 【Spark Summit East 2017】使用Spark进行时间序列分析

Typical time series analytics workflow:

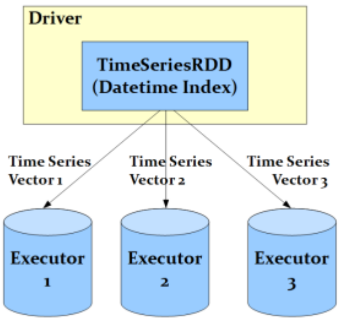



**Design Question #2**: How do we partition the multi-variate time series for distributed processing?

Across features, or across time?

## Current design

Assumption: a single time series must fit inside executory memory!




## Current design

Assumption: a single time series must fit inside executory memory!

```
TimeSeriesRDD (
    DatetimeIndex,
    RDD[(K, Vector)]
)

IrregularDatetimeIndex (
    Array[Long],      // Other limitation: Scala arrays = 2^32 elements
    java.time.ZoneId
)
```

【Spark Summit East 2017】使用Spark进行时间序列分析

# Future Improvements

- Creation of a new TimeSeriesRDD-like class that will be longitudinally (i.e. across time) partitioned rather than horizontally (i.e. across features).
- Keep both types of partitioning, on a case-by-case basis.

**Design Question #3**: How do we lag, difference, etc.?

Re-sampling, or index-preserving?

# Option #1: re-sampling

**Before**

| Irregular Time | y value at t | x value at t |
|---|---|---|
| 1:30:05 | 51.42 | 4.87 |
| 1:30:07.86 | 52.37 | 4.99 |
| 1:30:07.98 | 53.22 | 4.95 |
| 1:30:08.04 | 55.87 | 4.97 |
| 1:30:12 | 54.84 | 5.12 |
| 1:30:14 | 49.88 | 5.10 |

**After (1 second lag)**

| Uniform Time | y value at t | x value at (t – 1) |
|---|---|---|
| 1:30:06 | 51.42 | 4.87 |
| 1:30:07 | 51.42 | 4.87 |
| 1:30:08 | 53.22 | 4.87 |
| 1:30:09 | 55.87 | 4.95 |
| 1:30:10 | 55.87 | 4.97 |
| 1:30:11 | 55.87 | 4.97 |
| 1:30:12 | 54.84 | 4.97 |
| 1:30:13 | 54.84 | 5.12 |
| 1:30:14 | 49.88 | 5.12 |

# Option #2: Index preserving

**Before**

| Irregular Time | y value at t | x value at t |
|---|---|---|
| 1:30:05 | 51.42 | 4.87 |
| 1:30:07.86 | 52.37 | 4.99 |
| 1:30:07.98 | 53.22 | 4.95 |
| 1:30:08.04 | 55.87 | 4.97 |
| 1:30:12 | 54.84 | 5.12 |
| 1:30:14 | 49.88 | 5.10 |

**After (1 second lag)**

| Irregular Time | y value at t | x value at (t – 1) |
|---|---|---|
| 1:30:05 | 51.42 | N/A |
| 1:30:07.86 | 52.37 | 4.87 |
| 1:30:07.98 | 53.22 | 4.87 |
| 1:30:08.04 | 55.87 | 4.87 |
| 1:30:12 | 54.84 | 4.97 |
| 1:30:14 | 49.88 | 5.12 |

SPARK SUMMIT EAST 2017 云栖社区 yq.aliyun.com FAIMDATA

# Current functionality

- Option #1: resample() function for lagging/differencing by upsampling/downsampling.
  – Custom interpolation function (used when downsampling)

- Conceptual problems:
  – Information loss and duplication (downsampling)
  – Bloating (upsampling)

SPARK SUMMIT EAST 2017 云栖社区 yq.aliyun.com FAIMDATA

# Current functionality

- Option #2: functions to lag/difference irregular time series based on arbitrary time intervals. (preserves index)

- Same thing: custom interpolation function can be passed for when downsampling occurs.

SPARK SUMMIT EAST 2017 云栖社区 yq.aliyun.com FAIMDATA

## Overview of current API

SPARK SUMMIT EAST 2017 云栖社区 yq.aliyun.com FAIMDATA

## High-level objects

- TimeSeriesRDD
- TimeSeries
- TimeSeriesStatisticalTests
- TimeSeriesModel
- DatetimeIndex
- UnivariateTimeSeries

## TimeSeriesRDD

- collectAsTimeSeries
- filterStartingBefore, filterStartingAfter, slice
- filterByInstant
- quotients, differences, lags
- fill: fills NaNs by specified interpolation method (*linear, nearest, next, previous, spline, zero*)
- mapSeries
- seriesStats: min, max, average, std. deviation
- toInstants, toInstantsDataFrame
- resample
- rollSum, rollMean
- saveAsCsv, saveAsParquetDataFrame

## TimeSeriesStatisticalTests

- Stationarity tests:
  - Augmented Dickey-Fuller (adftest)
  - KPSS (kpsstest)
- Serial auto-correlation tests:
  - Durbin-Watson (dwtest)
  - Breusch-Godfrey (bgtest)
  - Ljung-Box (lbtest)
- Breusch-Pagan heteroskedasticity test (bptest)
- Newey-West variance estimator (neweyWestVarianceEstimator)

## TimeSeriesModel

- AR, ARIMA
- ARX, ARIMAX (i.e. with exogenous variables)
- Exponentially weighted moving average
- Holt-winters method (triple exp. smoothing)
- GARCH(1,1), ARGARCH(1,1,1)

【Spark Summit East 2017】使用Spark进行时间序列分析

# Others

- Java bindings
- Python bindings
- YAHOO financial data parser

云栖社区 yq.aliyun.com FAIMDATA

# Code example #1

| Time | Y | X |
|---|---|---|
| 12:45:01 | 3.45 | 25.0 |
| 12:46:02 | 4.45 | 30.0 |
| 12:46:58 | 3.45 | 40.0 |
| 12:47:45 | 3.00 | 35.0 |
| 12:48:05 | 4.00 | 45.0 |

Y is stationary
X is integrated of order 1

云栖社区 yq.aliyun.com FAIMDATA

# Code example #1

```scala
val ts = TimeSeriesRDD.timeSeriesRDDFromCsv("mydata.csv", sc)

val newIndex = ts.index.islice(1, ts.index.size)

val tsTransformed = ts.mapSeries(vec => {
  val result = TimeSeriesStatisticalTests.adftest(vec, 0, "c")
  if (result._2 > 0.05) differencesAtLag(vec, 1) else vec
}, newIndex).lags(2, Map(("y" -> true)))

val instantsAsLPs = tsTransformed.toInstants().map(row =>
  LabeledPoint(row._2(0), Vectors.dense(row._2.toArray.drop(1))))

val algo = new LassoWithSGD().setIntercept(true)
algo.optimizer.setRegParam(0.5)
val model = algo.run(instantsAsLPs)
```

云栖社区 yq.aliyun.com FAIMDATA

【Spark Summit East 2017】使用Spark进行时间序列分析

## Code example #1

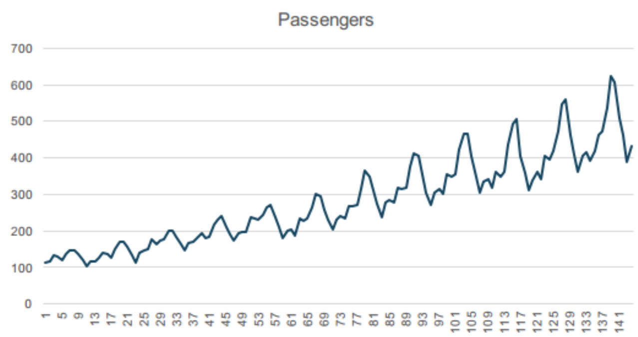
| Time | y | d(x) | Lag1(y) | Lag2(y) | Lag1(d(x)) | Lag2(d(x)) |
|---|---|---|---|---|---|---|
| 12:45:01 | 3.45 | | | | | |
| 12:46:02 | 4.45 | 5.0 | 3.45 | | | |
| 12:46:58 | 3.45 | 10.0 | 4.45 | 3.45 | 5.0 | |
| 12:47:45 | 3.00 | -5.0 | 3.45 | 4.45 | 10.0 | 5.0 |
| 12:48:05 | 4.00 | 10.0 | 3.00 | 3.45 | -5.0 | 10.0 |

## Code example #2

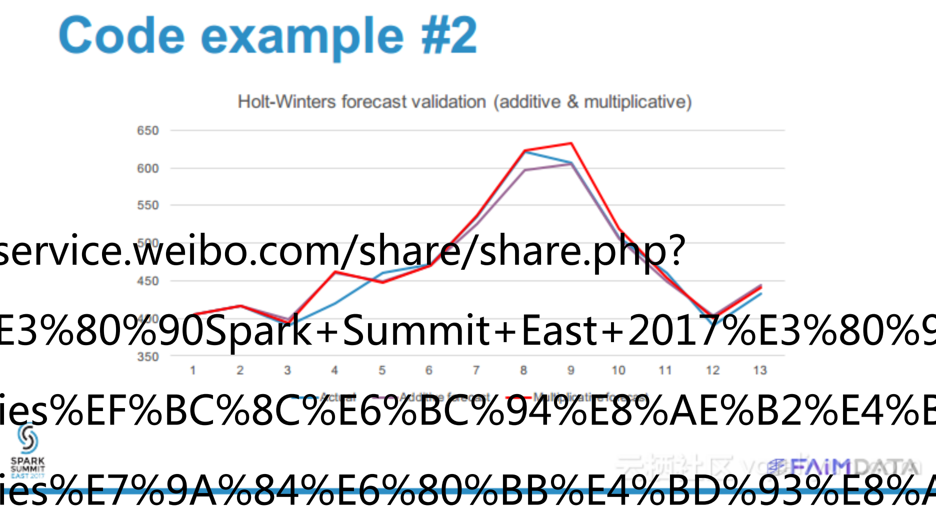
- We will use Holt-Winters to forecast some seasonal data.
- Holt-winters: exponential moving average applied to level, trend and seasonal component of the time series, then combined into global forecast.

$$
\begin{aligned}
l_x &= \alpha(y_x - s_{x-L}) + (1-\alpha)(l_{x-1} + b_{x-1}) && \text{level} \\
b_x &= \beta(l_x - l_{x-1}) + (1-\beta)b_{x-1} && \text{trend} \\
s_x &= \gamma(y_x - l_x) + (1-\gamma)s_{x-L} && \text{seasonal} \\
\hat{y}_{x+m} &= l_x + mb_x + s_{x-L+1+(m-1)mod(L)} && \text{forecast}
\end{aligned}
$$

## Code example #2



**【Spark Summit East 2017】使用Spark进行时间序列分析**


://service.weibo.com/share/share.php?
=%E3%80%90Spark+Summit+East+2017%E3%80%91%E4%BD%BF%E7%94
series%EF%BC%8C%E6%BC%94%E8%AE%B2%E4%B8%AD%E5%88%86%E4
series%E7%9A%84%E6%80%BB%E4%BD%93%E8%AE%BE%E8%AE%A1%EF


▸版权声明：本文内容由互联网用户自发贡献，版权归作者所有，本社区不拥有所有权，也不承担相关法律责任。如果您发现本社区中有涉嫌抄袭的内容，欢迎发送邮件至：yqgroup@service.aliyun.com (mailto:yqgroup@service.aliyun.com) 进行举报，并提供相关证据，一经查实，本社区将立刻删除涉嫌侵权内容。

用云栖社区APP，舒服~

【云栖快讯】直播推荐——现在报名3月12日编程语言系列讲座，与行业资深专家一起学习Python、C++、JavaScript、Java！还可在活动页面领取红包，百分百中奖哦！ 详情请点击 (https://yq.aliyun.com/promotion/543)

评论 (0) 点赞 (1) 收藏 (1)

# 【Spark Summit East 2017】使用Spark进行时间序列分析

分享到:

(http://service.weibo.com/sh
title=%E3%80%90Spark+Su
timeseries%EF%BC%8C%E6
timeseries%E7%9A%84%E6

上一篇：【Spark Summit East 2017】教会Spark集群弹性管理… 下一篇：【Spark Summit East 2017】Bulletproof Jobs：大规…

## 相关文章

满满的技术干货！Spark顶级会议Apache Spar… (/articles/69815) 【PDF大放送】Spark&Hadoop Summit精… (/articles/72207)

【Spark Summit East 2017】不再有“… (/articles/70344) 【Spark Summit East 2017】实时业务… (/articles/70334)

【Spark Summit East 2017】提升Py… (/articles/70367) 【Spark Summit East 2017】使用Sp… (/articles/70311)

【Spark Summit East 2017】使用Sp… (/articles/70306) 【Spark Summit East 2017】使用Sp… (/articles/70345)

【Spark Summit East 2017】加速云上… (/articles/70383) 【Spark Summit East 2017】R与Sp… (/articles/70424)

## 网友评论

登录后可评论，请 登录 (https://account.aliyun.com/login/login.htm?

**热点导航** **用户关注** **更多推荐**

闲时流量包 (https://promotion.aliyun.com/ntms/act/flowbagidle.html)云计算 (https://www.aliyun.com/)网络安全 (https://market.aliyun.com/security)互联网架构 (https://www.aliyun.com/aliwar
ECS升级配置 (https://yq.aliyun.com/ask/53742)物联网 (https://www.aliyun.com/product/iot)教程 (https://edu.aliyun.com/jiaocheng)PHP (https://yq.aliyun.com/php)
自动化测试 (https://bbs.aliyun.com/read/301499.html)解决方案 (https://www.aliyun.com/solution/all)linux命令 (https://yq.aliyun.com/articles/34777)云服务 (https://www.aliyun.com/ )
JavaScript 函数 (https://yq.aliyun.com/articles/92145)服务器监控 (https://yq.aliyun.com/articles/48786)Python语言 (https://yq.aliyun.com/roundtable/56407)移动数据分析 (https://www.aliyun
用户体验 (https://yq.aliyun.com/articles/132294)云数据库Rds (https://help.aliyun.com/product/26090.html)负载均衡 (https://www.aliyun.com/product/slb/)域名注册 (https://wanwang.aliyun.c
Whois查询 (https://whois.aliyun.com)数据可视化 (https://help.aliyun.com/product/43570.html)ICP备案查询 (https://beian.aliyun.com)主题地图 (https://yq.aliyun.com/zt)阿里云大学 (https://ed
cn域名 (https://wanwang.aliyun.com/domain/cn/)Js (https://yq.aliyun.com/jsarticle)Mysql (https://yq.aliyun.com/sqlarticle)移动站 (https://m.aliyun.com/yunqi/)IT论坛 (https://bbs.aliyun.com/)
企业邮箱 (https://mail.aliyun.com/)签名文件 (https://www.aliyun.com/jiaocheng/1075.html)

关于我们 (//www.aliyun.com/about) 法律声明及隐私权政策 (http://terms.aliyun.com/legal-agreement/terms/suit_bu1_ali_cloud/suit_bu1_ali_cloud201710161525_98396.html) 廉正举报 (https://jubao.alibaba.com/index.html?site=ALIYUN) (//www.aliyun.com/links)

阿里巴巴集团 (http://www.alibabagroup.com/cn/global/home) 淘宝网 (//www.taobao.com/) 天猫 (//www.tmall.com/) 聚划算 (//ju.taobao.com/) 全球速卖通 (//www.aliexpre

阿里巴巴国际交易市场 (//www.alibaba.com/) 1688 (//www.1688.com/) 阿里妈妈 (//www.alimama.com/index.htm) 飞猪 (//www.alitrip.com/) 阿里云计算 (//www.aliyun.com

YunOS (//www.yunos.com/) 阿里通信 (//aliqin.tmall.com/) 万网 (//wanwang.aliyun.com/) 高德 (http://www.autonavi.com/) UC (http://www.uc.cn/) 友盟 (//www.umeng.co

虾米 (//www.xiami.com/) 阿里星球 (//www.alibabaplanet.com) 来往 (//www.laiwang.com/) 钉钉 (//www.dingtalk.com/?lwfrom=20150205111943449) 支付宝 (https://www.a

© 2009-2018 Aliyun.com 版权所有 ICP证：浙B2-20080101

(http://idinfo.zjaic.gov.cn/bscx.do?method=hddoc&id=33010800001867)

浙公网安备 33010002000099号(http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=33010002000099)