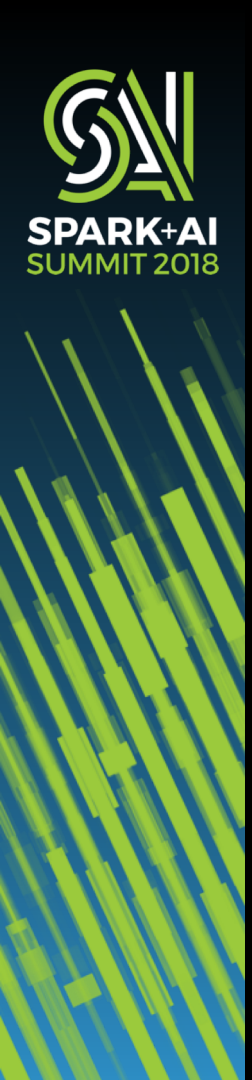


Apache Spark and Machine Learning Boosts Revenue Growth for Online Retailers

Ruifeng Zheng, JD.COM

Yanbo Liang, Hortonworks

About us

Ruifeng Zheng ruifengz@foxmail.com

- Senior Software Engineer in Intelligent Advertising Lab at JD.COM
- Apache Spark, Scikit-Learn & XGBoost contributor
- SparkLibFM & SparkGBM Author

Yanbo Liang ybliang8@gmail.com

- Staff Software Engineer at Hortonworks
- Apache Spark PMC member
- Tensorflow & XGBoost contributor

Outline

- What are the problems?
- How we solve it?
- The lessons learned.
- Where is the gap?
- Enhancements
  - ALS with warm start
  - SparkGBM – a new GBM impl atop Spark
- Future work

About JD.com & Wiwin

JD.com

- China's largest online retailer
- China's largest e-commerce delivery system
- 300+ million active users
- Billions of SKUs on shelves, in thousands of categories

WiWin Team in Business Growth Dept.

- Supply Data-Mining Services for Top Brands

Business scenarios

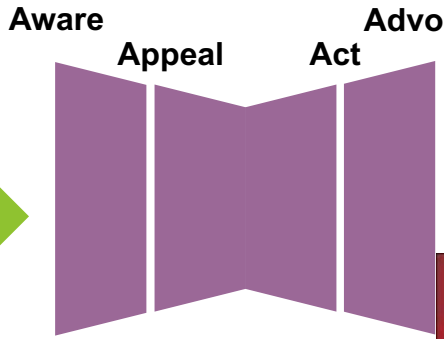
- User Segmentation
- Cross Selling
- Purchase Prediction

User Segmentation

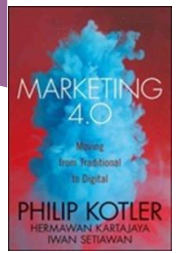
Demand: Help Brands to measure marketing campaigns (beyond ROI and GMV)

Impression, Click, View, Search, Ask, Follow, Order, Comment, Cart, etc

Input



4A Model



4A status of each user

User Segmentation

V1: RDD only

V2: DataFrame + RDD(only used in complexed operations)

~3.2x speedup
save 40% memory footprint

Cross-Selling

Demand: Help Brands to find potential co-operators among millions of brands

Cross-Selling

$$Support = \frac{frq(X,Y)}{N}$$

$$Rule: X \Rightarrow Y$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

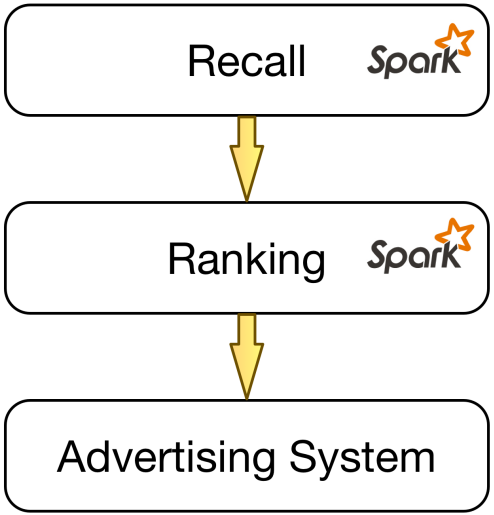
Measure how many more times X and Y occur together than expected

Flexible pattern #SPARK-13385

rules like {A,B,C} -> {X,Y}

Purchase Prediction

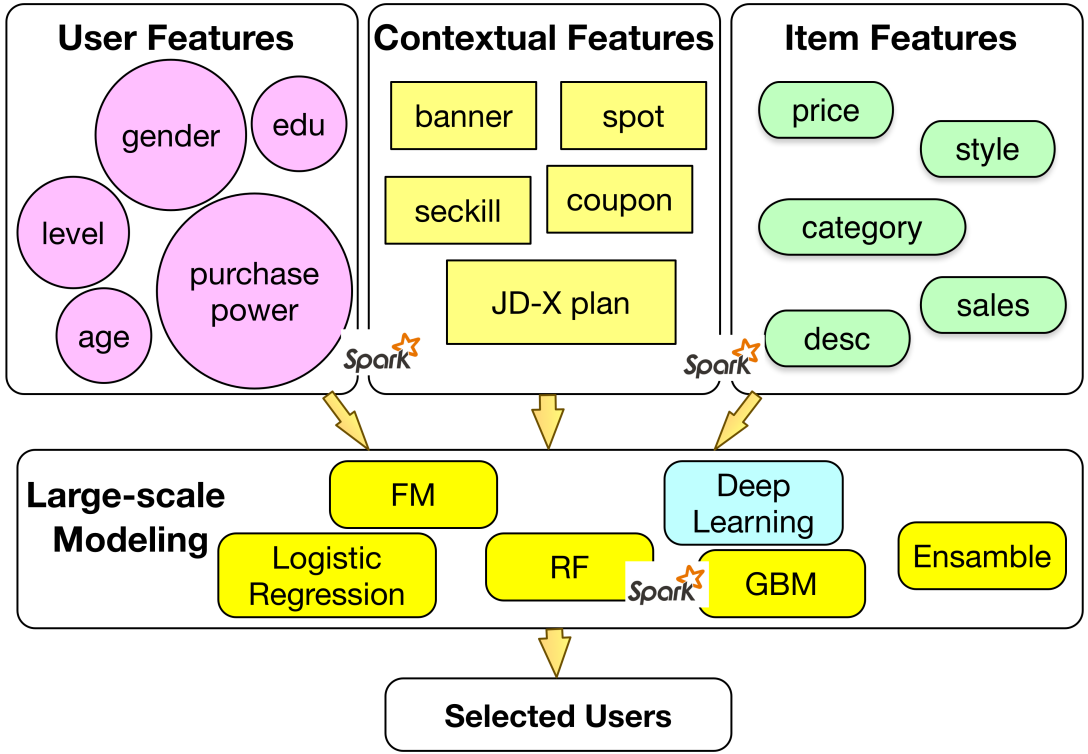
Demand: Help Brands to better target potential users



Different from tradition recommendation:

- For each user, select several items
- For several items, select millions of users

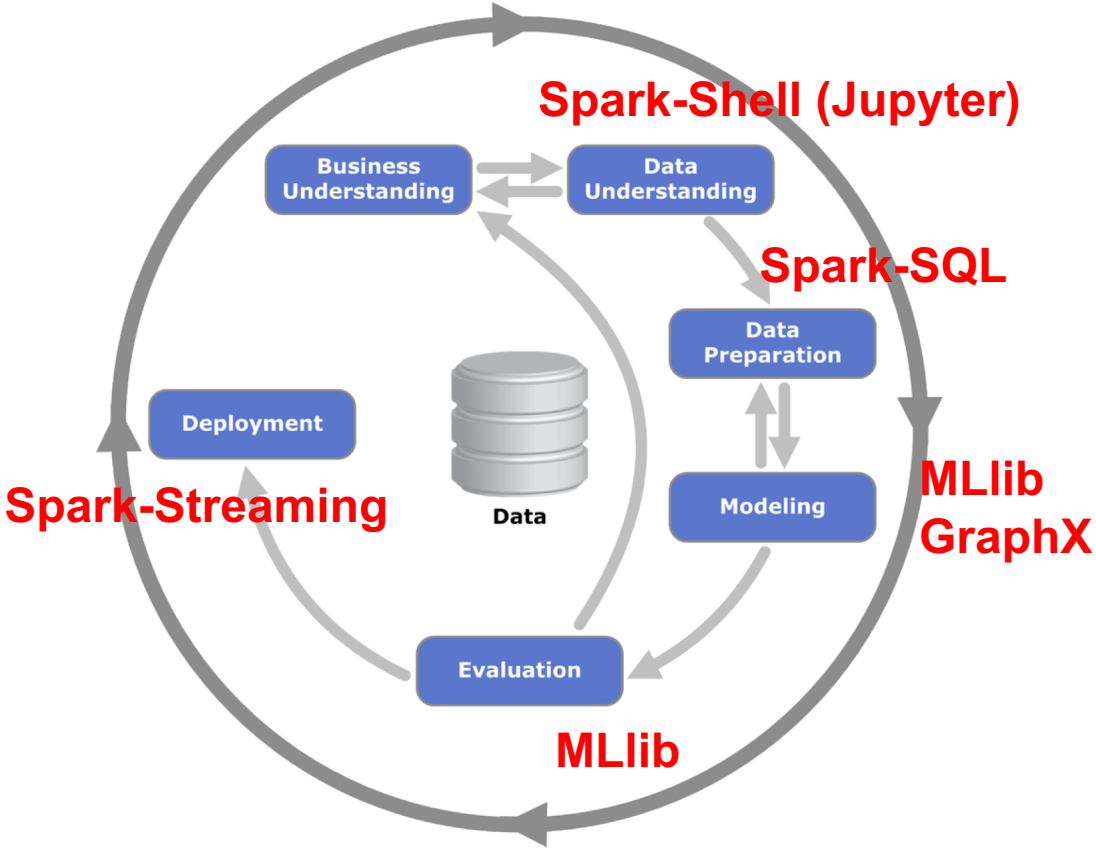
Purchase Prediction - Ranking



Pipeline

CRISP-DM

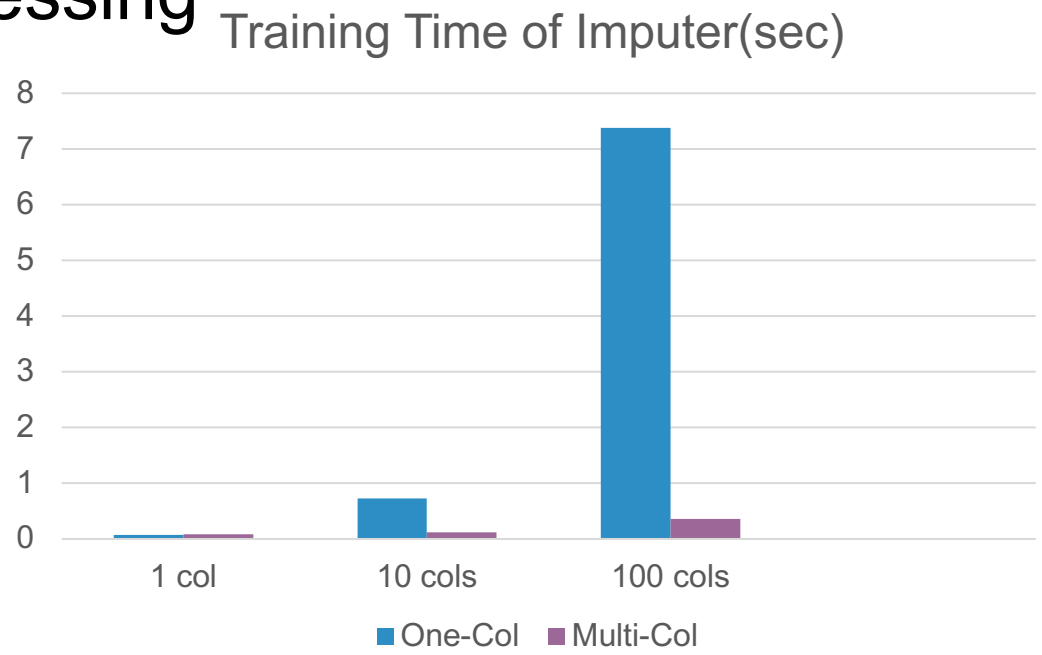
In-house data processing toolchain



Lessons Learned - 1

Multi-Column processing

- Imputer
  #SPARK-21690
- ApproxQuantile
  #SPARK-14352
- Bucketizer
  #SPARK-22797



Lessons Learned - 2

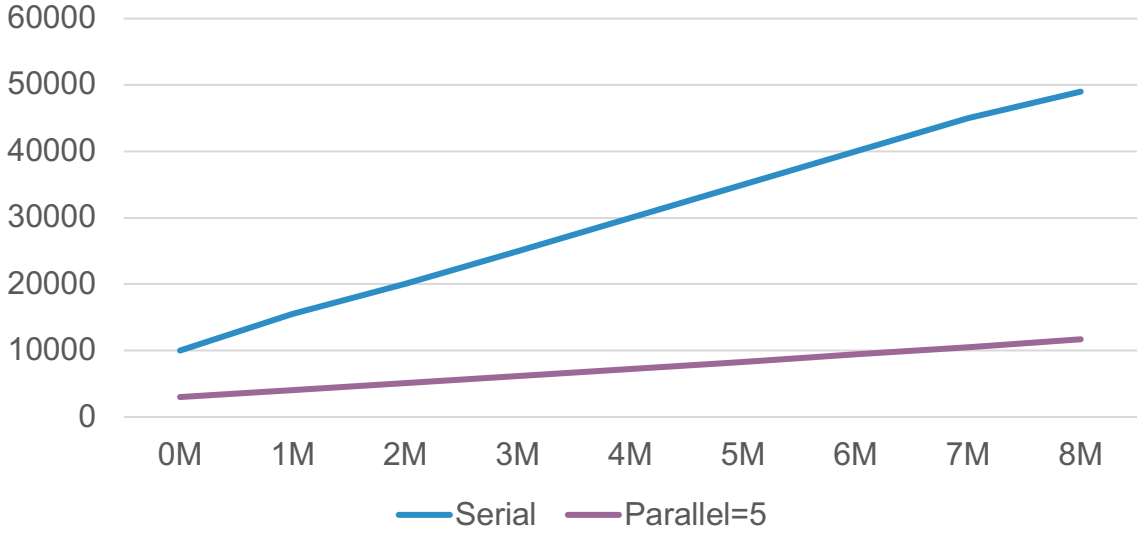
RDD & DataFrame are Complementary

ETL and data transformation -> DataFrame

Complex logic containing lots of aggregation -> RDD

Lessons Learned - 3

Parallelized Cross-Validation



https://bryancutler.github.io/cv-parallel/

GAP

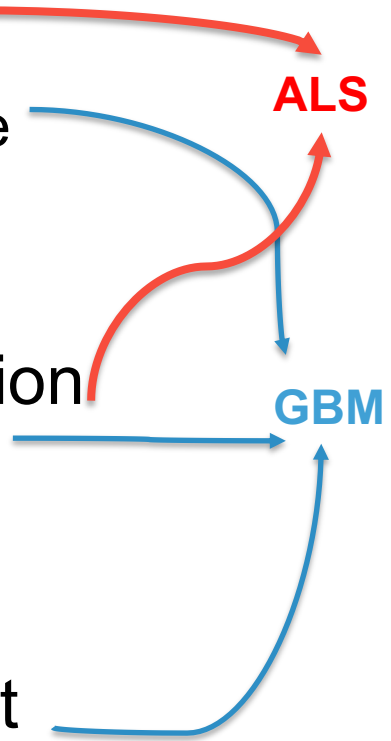
Warm Start

- Resume training
- Accelerate convergence
- Stable solution

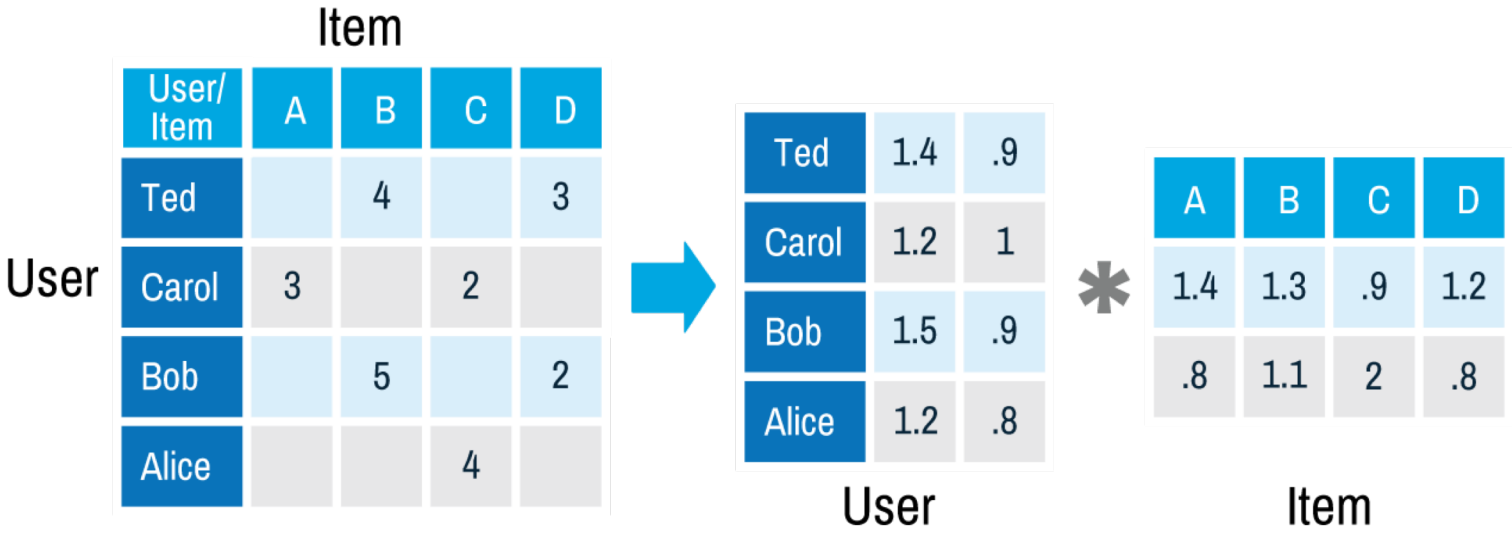
Callback after each iteration

- Early stop
- Model checkpoint

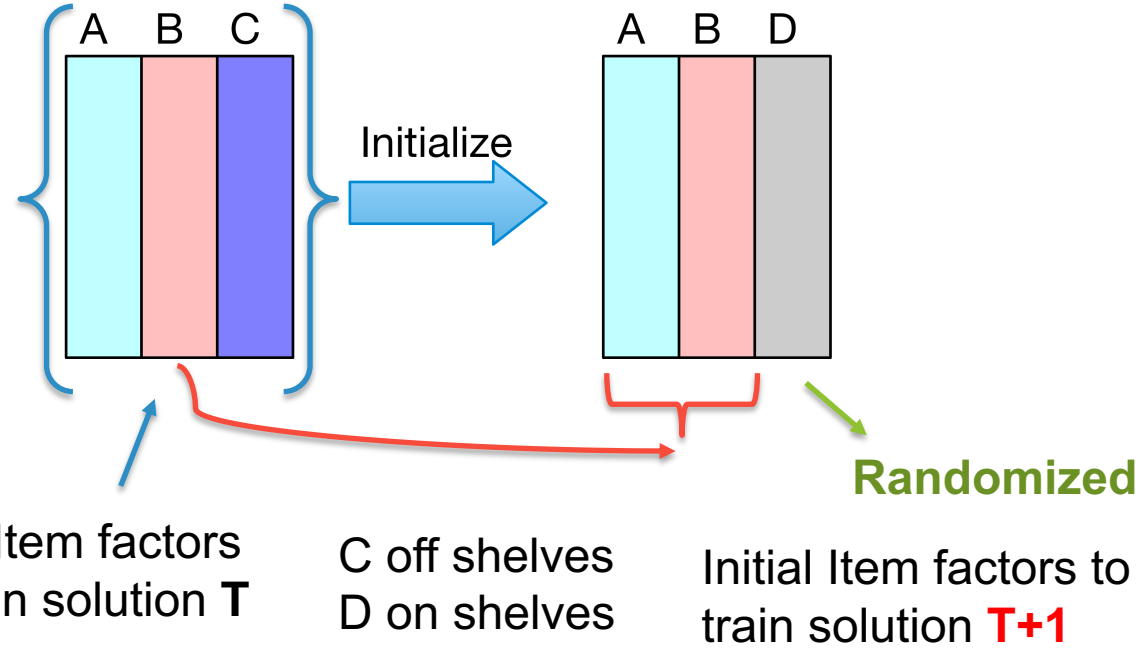
Compact Numeric Format



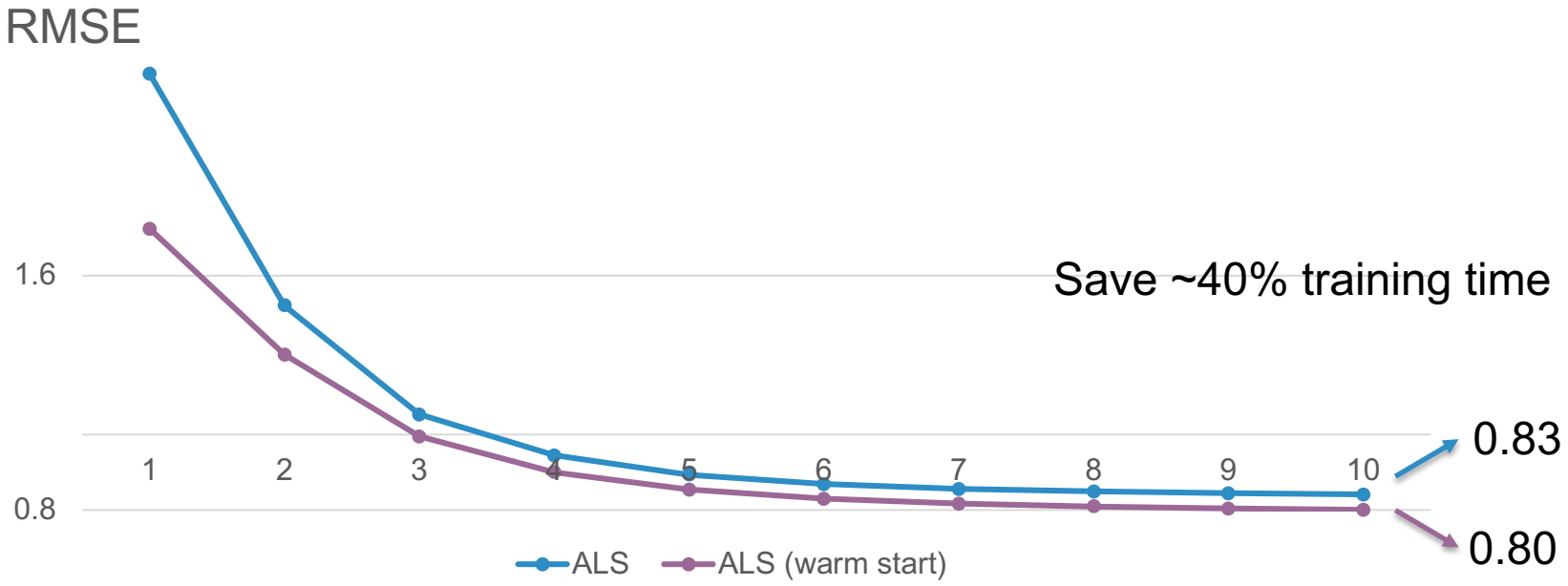
ALS


Item
User
User/ Item
A
B
C
D
Ted
4
3
Carol
3
2
Bob
5
2
Alice
4
Ted
1.4
.9
Carol
1.2
1
Bob
1.5
.9
Alice
1.2
.8
User
A
B
C
D
1.4
1.3
.9
1.2
.8
1.1
2
.8
Item

ALS – Warm start



ALS – Warm start



GBM - Life is short, you need GBM

Objective in t-th Iteration:

$$Obj^{t}(\theta) = \sum_{i=1}^{t-1} L\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_t)$$

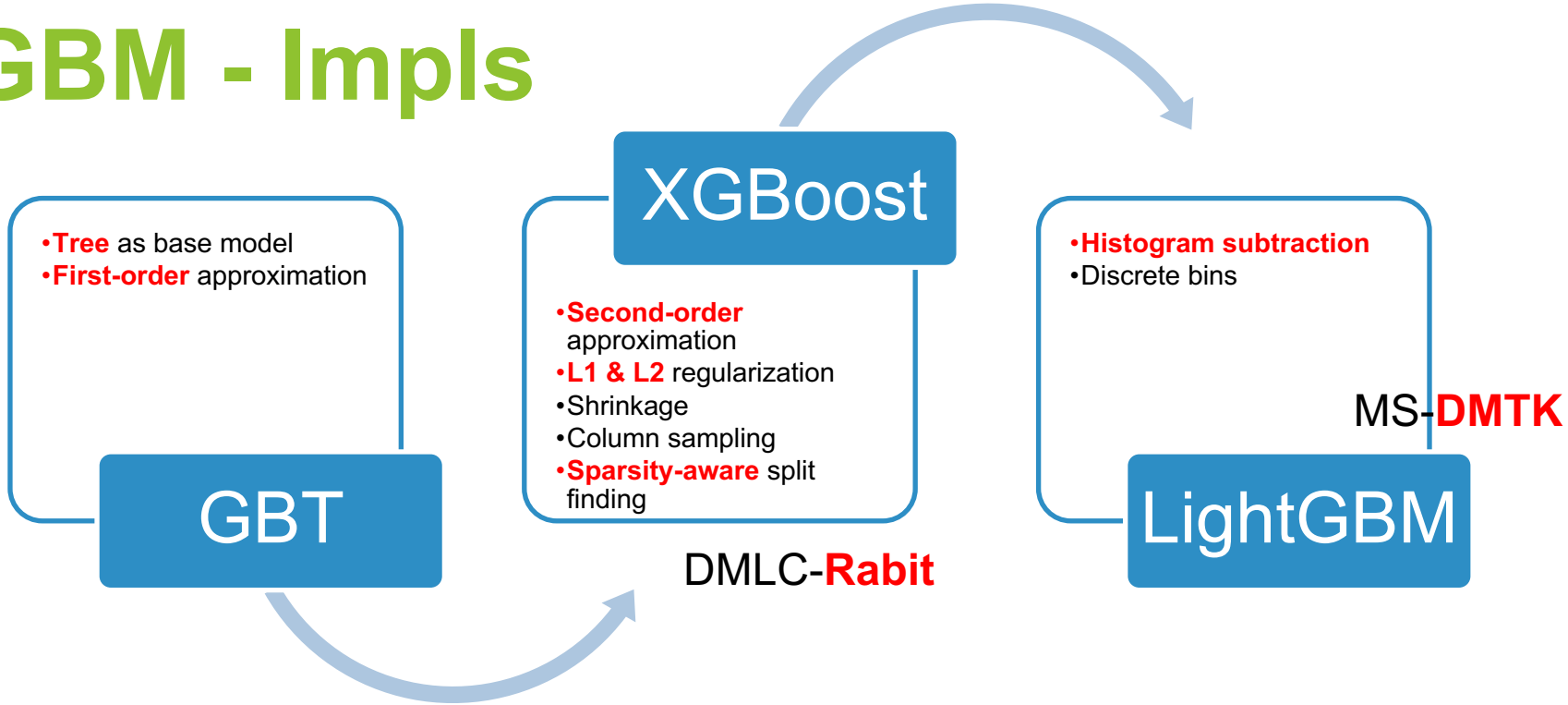
training loss: how well model fit on training data

previous prediction

base model to be added in Iteration t

regularization: control model complexity

GBM - Impls



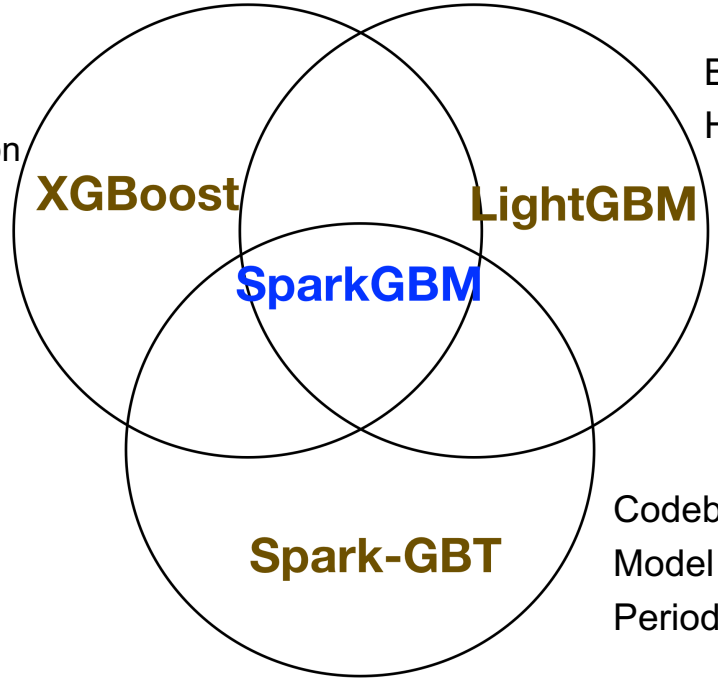
Dedicated ML frameworks result in extra costs in Deployment, Maintenance & Monitoring

SparkGBM

https://github.com/zhengruifeng/SparkGBM

To be a scalable and efficient GBM atop Spark

Second-order approximation
L1 & L2 regularization
Shrinkage
Column sampling
Sparsity-aware



Binned data
Histogram subtraction

Codebase legacy:
Model save/load
Periodic checkpointer
…

SparkGBM - Features

Compatible with MLLib pipeline

Warm start

Early stop

User-defined functions (RDD only)

- Objection
- Evaluation
- Callback: Early stopping, Model checkpoint

SparkGBM – API 1

GBMRegressor & GBMClassifier

```
val gbmr = new GBMRegressor
gbmr.setBoostType("dart")
        .setDropRate(0.1)
        .setObjectiveFunc("square")
        .setRegLambda(0.5)
        .setRegAlpha(0.1)
        .setEvaluateFunc(Array("rmse", "mae"))
        .setEarlyStopIters(10)
        .setInitialModelPath(path)
```

Gradient boosting & DART

Objective

Regularization

Early stop

Warm start

SparkGBM – API 2

GBMRegressionModel & GBMClassificationModel

```
val model1 = gbmr.fit(train)
```

Train without validation, early stop is **disabled**

```
val model2 = gbmr.fit(train, test)
```

Train with validation, early stop is **enabled**

```
model2.setFirstTrees(5)
```

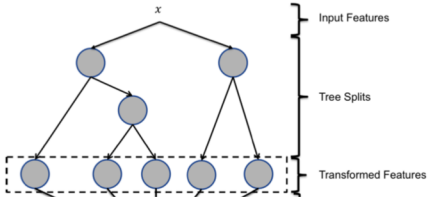
Using first 5 trees for following computation

```
model2.transform(test)
```

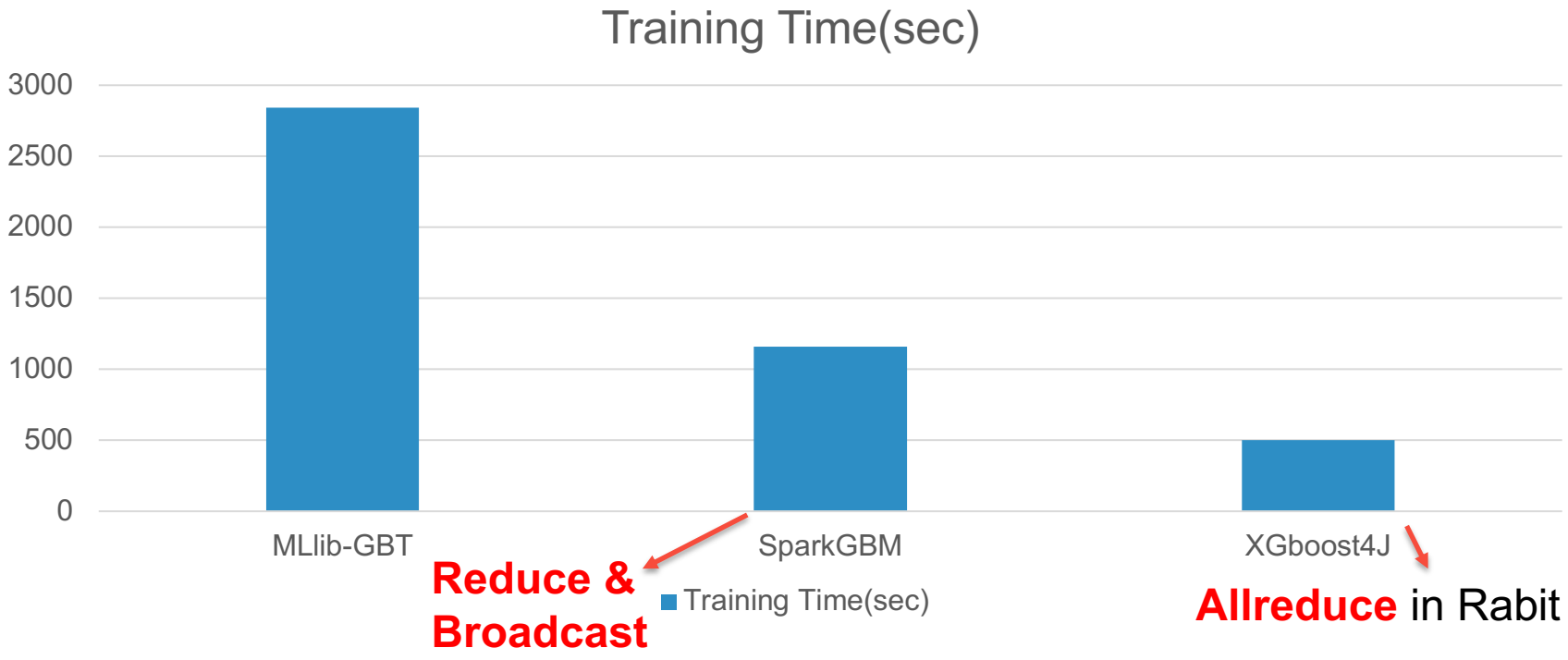
Prediction

```
model2.setEnableOneHot(true)
model2.leaf(test)
```

Feature transformation by index of leaf/path



SparkGBM – Performance



Future work

- Warm start in other algorithms
  - Use K-Means to initialize GMM
- ALS enhancements
  - Improve the solution stability
- SparkGBM enhancements
  - Add features from XGBoost & LightGBM, i.e. softmax to support multi-class classification

Thank you!