机器学习助力精准营销广告

CherryAI
徐煌

About Me

▶ 徐煌

1999 2003 2005 2009 2011 2016

目录

CONTENTS

The Bussiness

广告主	广告代理商	广告展示平台	最终用户

唯品会
全球精选 正品特卖

criteo

Google

晶赞科技
ZAMPLUS

YOYi 悠易
多屏程序化购买引领者

Only Big Whales

头条

Google

Tanx
阿里妈妈旗下产品

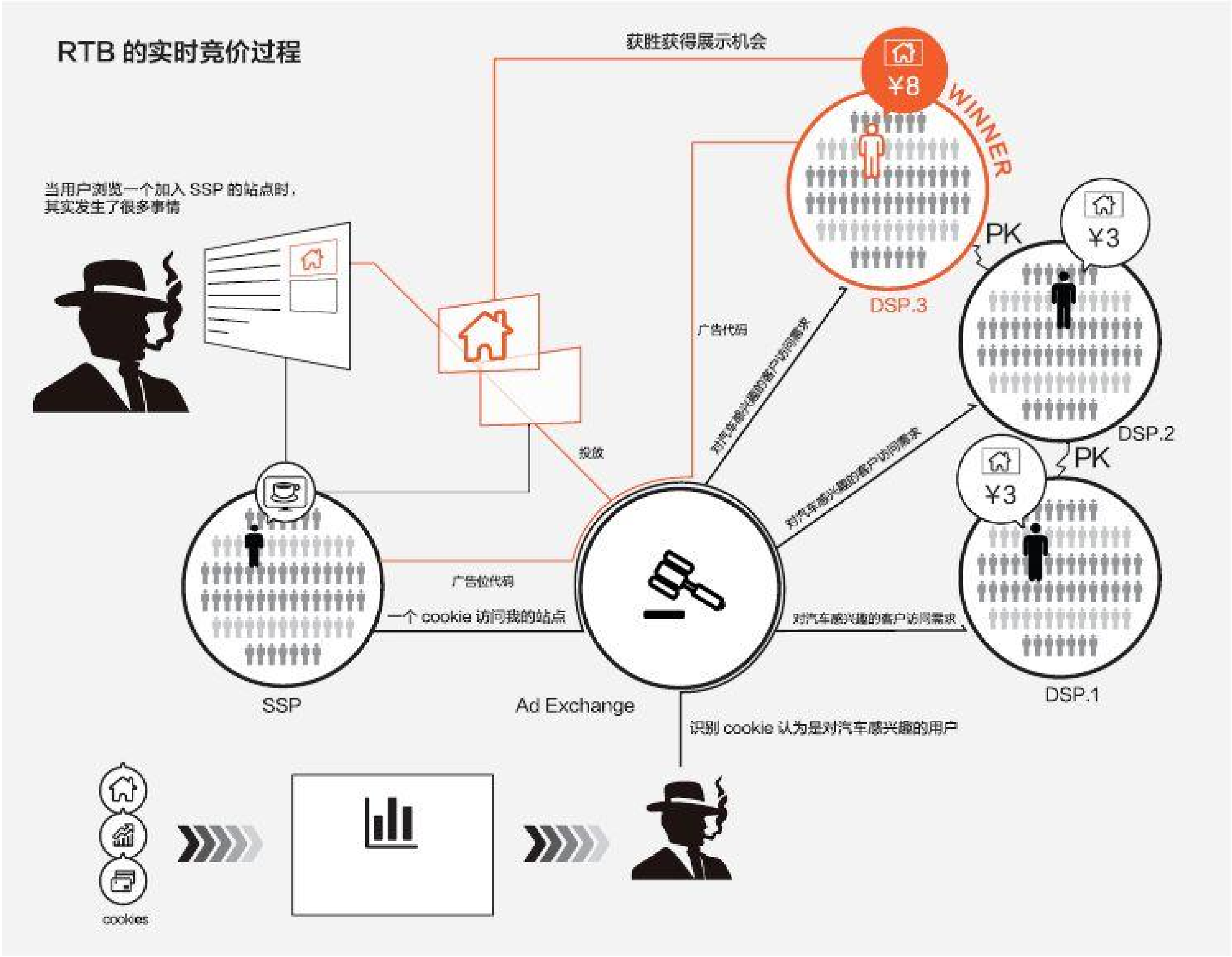
facebook

Baidu百度
百度一下 你就知道

Tencent腾讯

Buy options

今日头条 投放管理平台
设置广告受众
复制已有受众人群
自定义人群包
不限
定向人群包
排除人群包
地域
不限
省市
区县
商
性别
不限
男
女
年龄
不限
指定年龄段
兴趣分类
不限
添加兴趣
兴趣分类
全选
游戏
休闲时间
跑酷竞速
宝石消除
网络游戏
动作射击
扑克棋牌
兴趣关键词
不限
自选关键词
app行为定向
不限
按分类
按app
用户首次激活时间
不限
指定时间段



RTB 的实时竞价过程
获胜获得展示机会
¥8
WINNER
当用户浏览一个加入 SSP 的站点时，
其实发生了很多事情
PK
¥3
DSP.3
广告代码
DSP.2
PK
¥3
投放
广告位代码
一个 cookie 访问我的站点
对汽车感兴趣的客户访问需求
SSP
Ad Exchange
DSP.1
识别 cookie 认为是对汽车感兴趣的用户
cookies

Problems for Advertisers

- Audience?
- Price to buy Ads?
- Content that shows to the audience?
- When?
- ...

Performance Advertising

« The right ad at the right time to the right user »

6ms

How we earn money?

- Clients pay us per click, sale etc.
- We buys advertisements from publishers (google, facebook, etc.) in cost of displays.

- We earns the difference: Click * Cost per click - Cost of displays

目录

CONTENTS

ML is everywhere

- We use ML for:
  - Bidding
  - Campaign selection
  - Look&Feel optimization
  - Product recommendation

Campaign selection

- Choose the best client for current user

...

- Choose max estimated reward

Bidding

- Estimate the real value of a display
  - The estimated value (estimated cost per display) could be varied for different bussiness model (CPC/CRO/COS/Target COS)

Example: prediction CPM based on CPC in bidding

$$Buy\ ?\ \mathbb{E}[CPM] > CPM$$

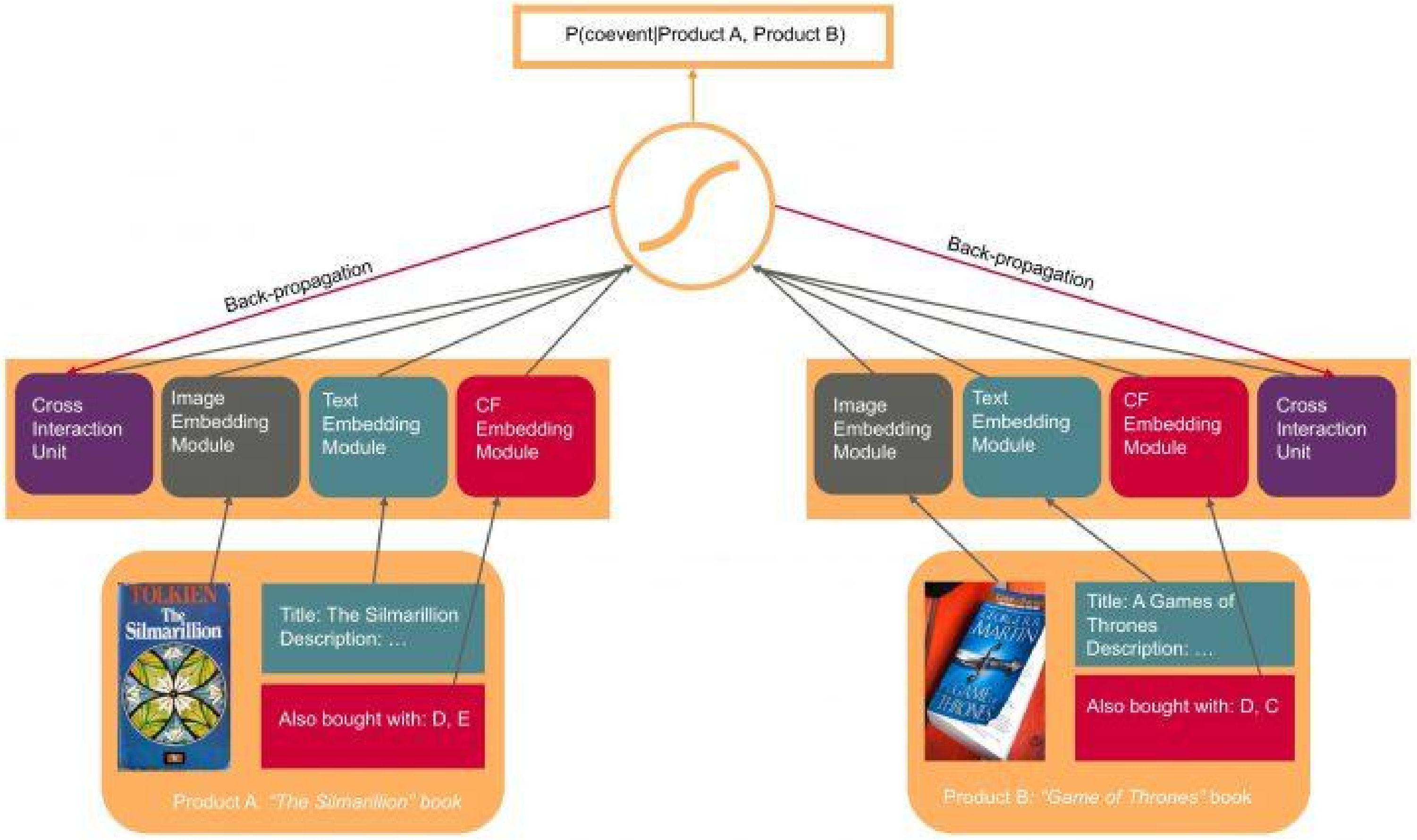
$$\mathbb{E}[CPM] = \mathbb{E}[NbClicks] * \text{CPC}$$

Recommendation

- Choose the best (Click Rate, Conversion Rate, Estimated Sales Amount) products to show in the banner

Content2vec: Unified product representation for recommender systems

- Collaborative Filtering and others informations
- Reach a unified product representation that gathers all information available on the products to enable us to do better recommendations.



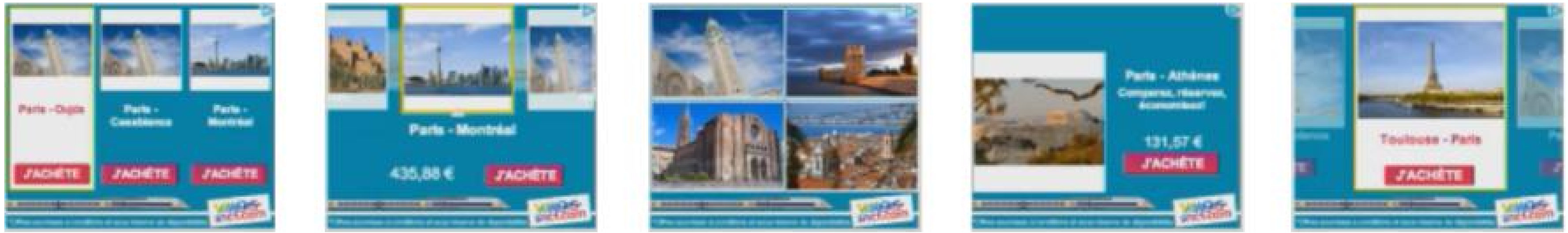
Contextual Recurrent Neural Networks for Recommendation

- RNN with context information

Dynamic rendering optimization

- Choose the look&feel of the banner

Layout:

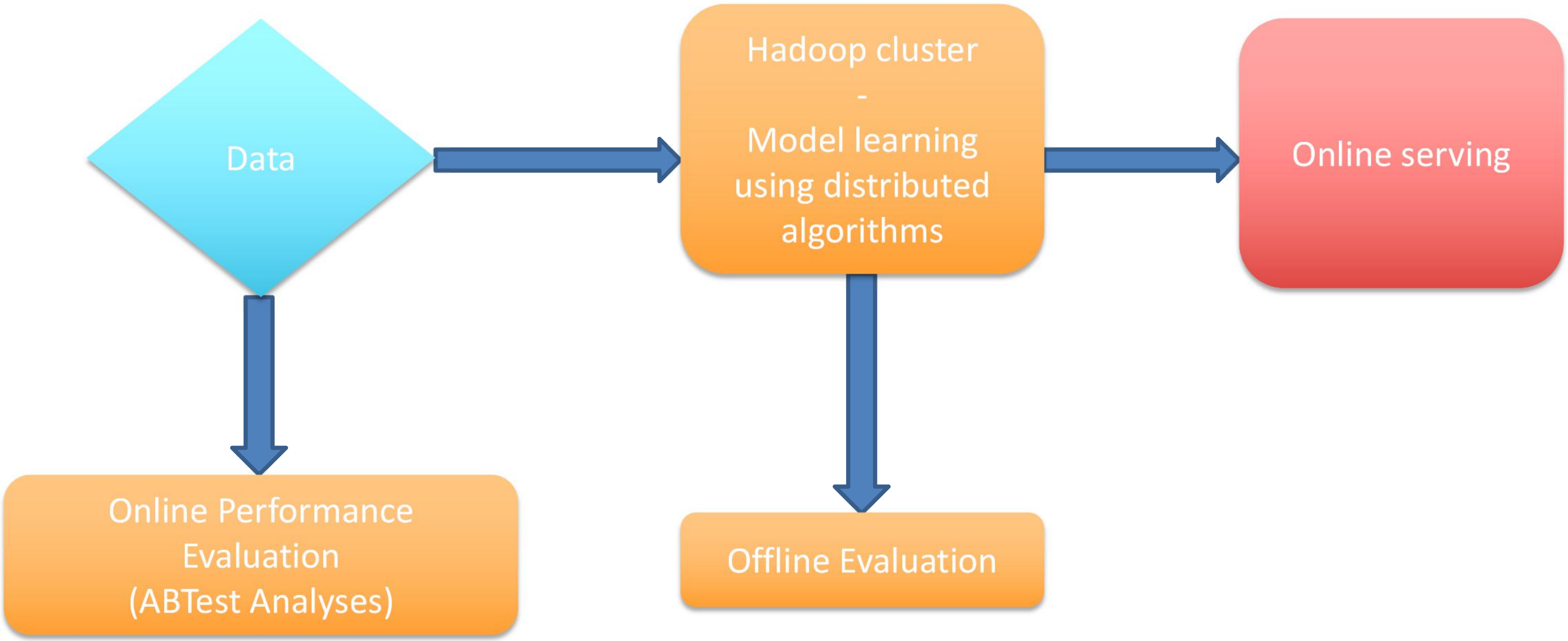


ColorSet:

目录

CONTENTS

Life Cycle


Data
Hadoop cluster
-
Model learning using distributed algorithms
Online serving
Online Performance Evaluation
(ABTest Analyses)
Offline Evaluation

Data

- Stored on Hadoop Distributed File System
- Raw data:
  - Compressed json
  - Different data sources: displays, clicks, sales, etc..
- Refined data:
  - Produced by Hadoop jobs (cascading, scalding)
  - Combine different data types
  - Exported in Parquet (column based) to accelerate reading

Offline Evaluation

- An internal tool that replays Prod traffic from logs with different prediction models
- Target:
  - Evaluation of new models offline before going to AbTest
  - Advance investigation of production models

Offline Evaluation: In metrics we trust

My ABTest: +10% RevExTac on the first 2 hours
Could we trust this improvement or is it just noise?

- Computing confidence interval:

Offline Evaluation: Simulation

- As prediction will impact bidding, so the data we receive, simulation of acquired data should be done.
- Using counterfactual estimators
- Bid with a gaussian bias

Online Serving

- 3 Billion displays per day

- All predictions don’t lead to a display

- More than 1000 billions predictions made every day !

- Some displays need up to 400 call to prediction API
- => must be very fast (50µs)

Online Performance Evaluation (ABTest analyses)

- All Product changes are validated by an AbTest (performance is everything)
  - Realtime monitoring: to secure AbTest in realtime
  - AbTest analysis Framework: to validate AbTest with deep insight and confidential interval

AbTest analysis Framework

Realtime monitoring

Lesson learned

- Quality data is important
- Use offline tests to tune models
- Use abtest to secure and evaluate changes

目录

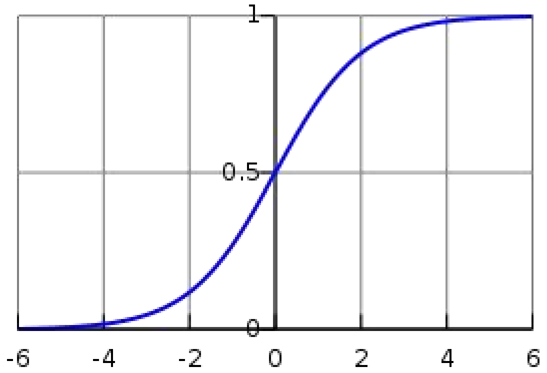
CONTENTS

Learning: Click prediction modelling

$$\mathbb{E}[NbClicks] = \mathbb{P}\{Click\}\mathbb{E}[NbClicks|Click]$$

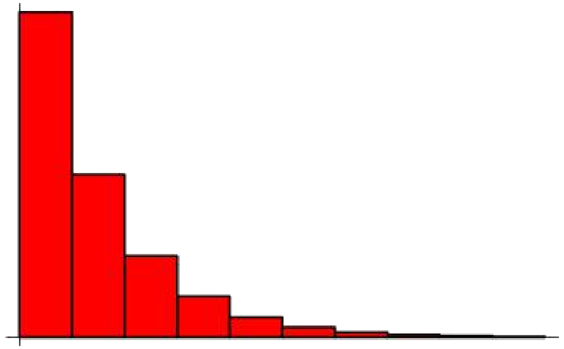
$$= \frac{1}{1+e^{-<w.x>}}$$

(logistic)



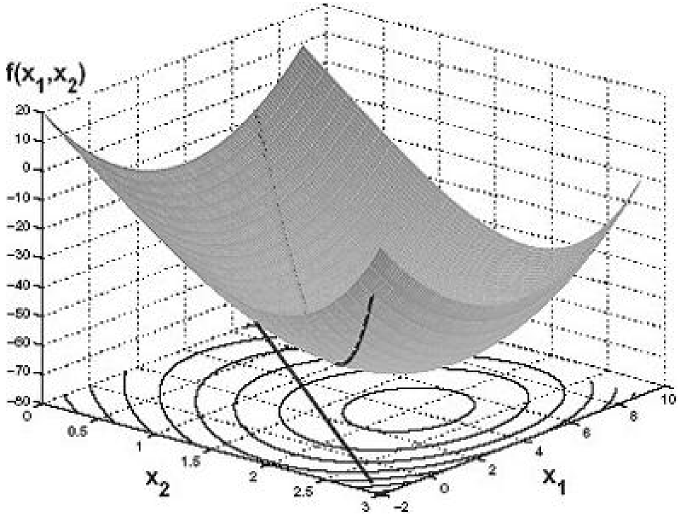
$$=1 + e^{-<w.x>}$$

(geometric)

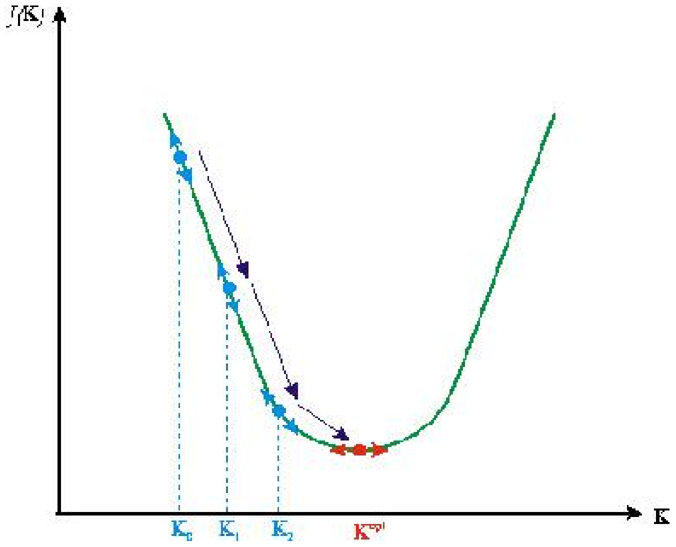


Learning: Logistic Regression

- The output model is a vector of weights : double[]



Convex Optimisation



Solvable with iterative Gradient Descent Algorithms (L-BFGS)

Fast Prediction at runtime

Learning: Need for scale

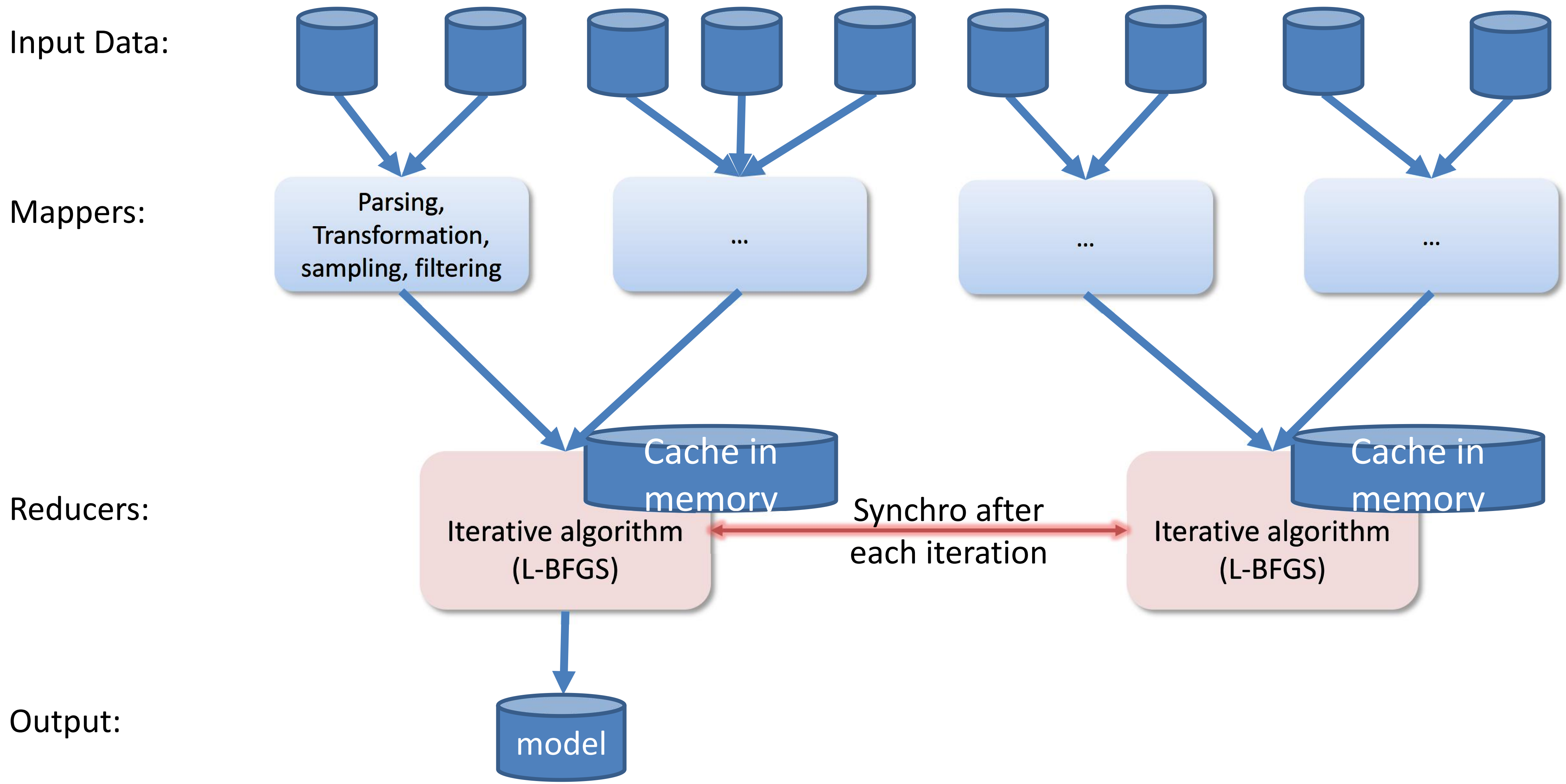
Learning a model (click prediction):

- Several days of data
- Billions of samples (after sampling)
- Millions of features

... and we have ~200 of them
(click/sales/... x DCs x ABTests)...

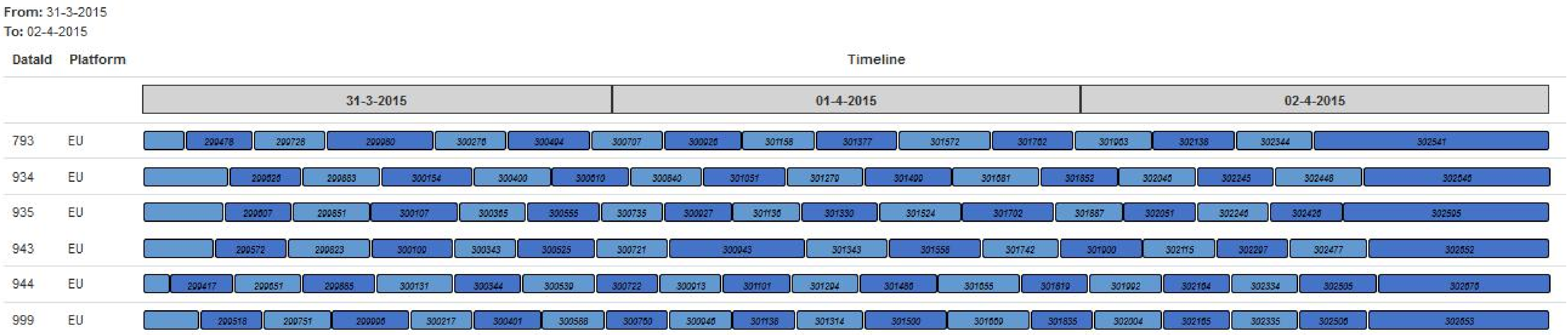
.. and we want to refresh as much as possible..

Learning with hadoop



Learning: Some numbers

- ~ 1300 models/day
- Ingesting 596 TB/day
- Consuming 6310 CPU day/day
- Learning time: [10min; 3h]
- Refresh rate: [3h; 6h]



Learning: Lesson learned

- Balance your data
- Hash
- Tradeoff: reactive vs stable

Thanks!

Q & A