Machine Learning in the Cloud with Spark

R212 Data Centric Systems and Networking
Open Source Project Study

by Haikal Pribadi

Machine Learning in the Cloud with Spark

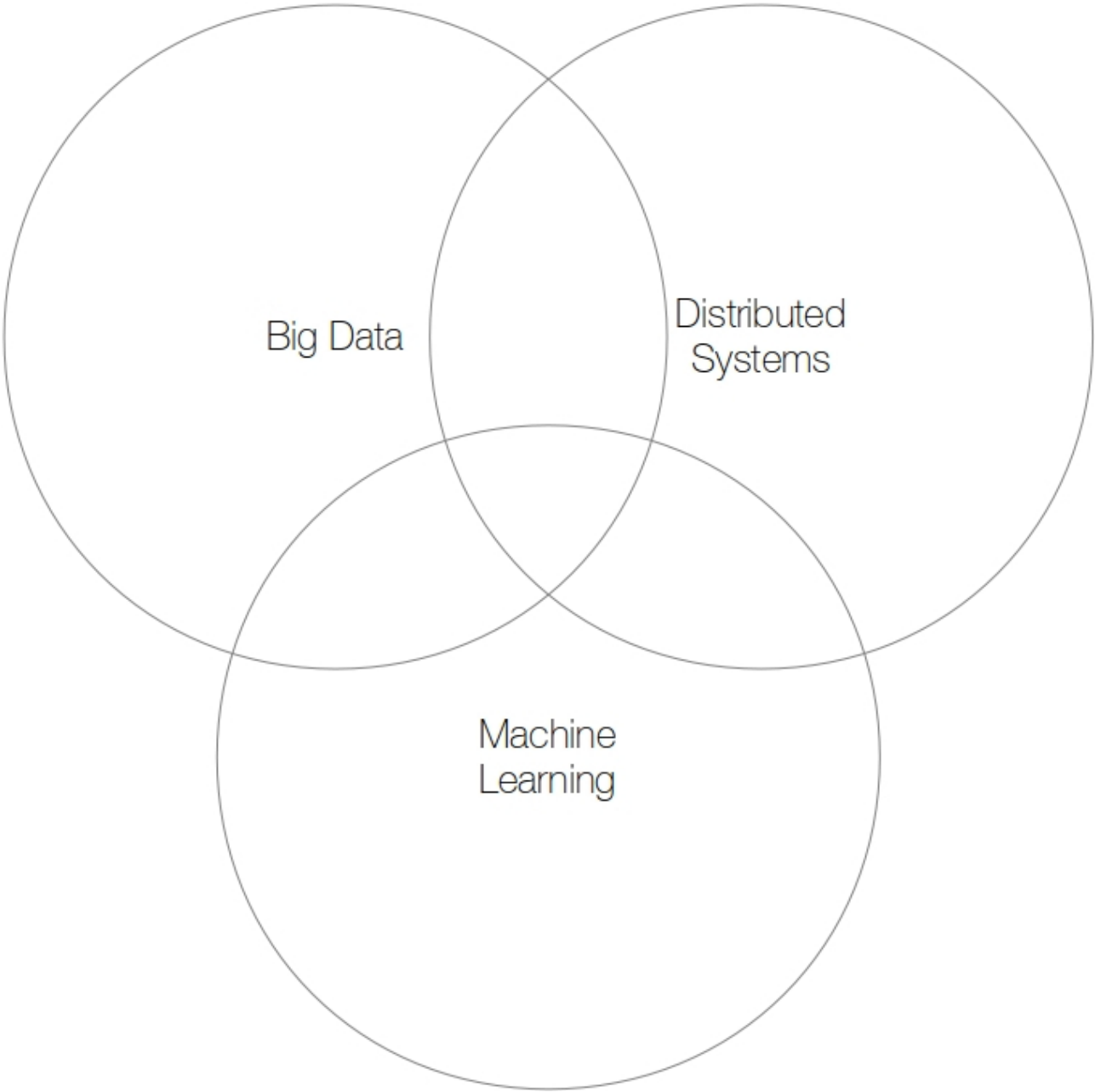
- Problem domain
- Motivation and Contribution
- Project Goal
- Project Evaluation

Converging Trends

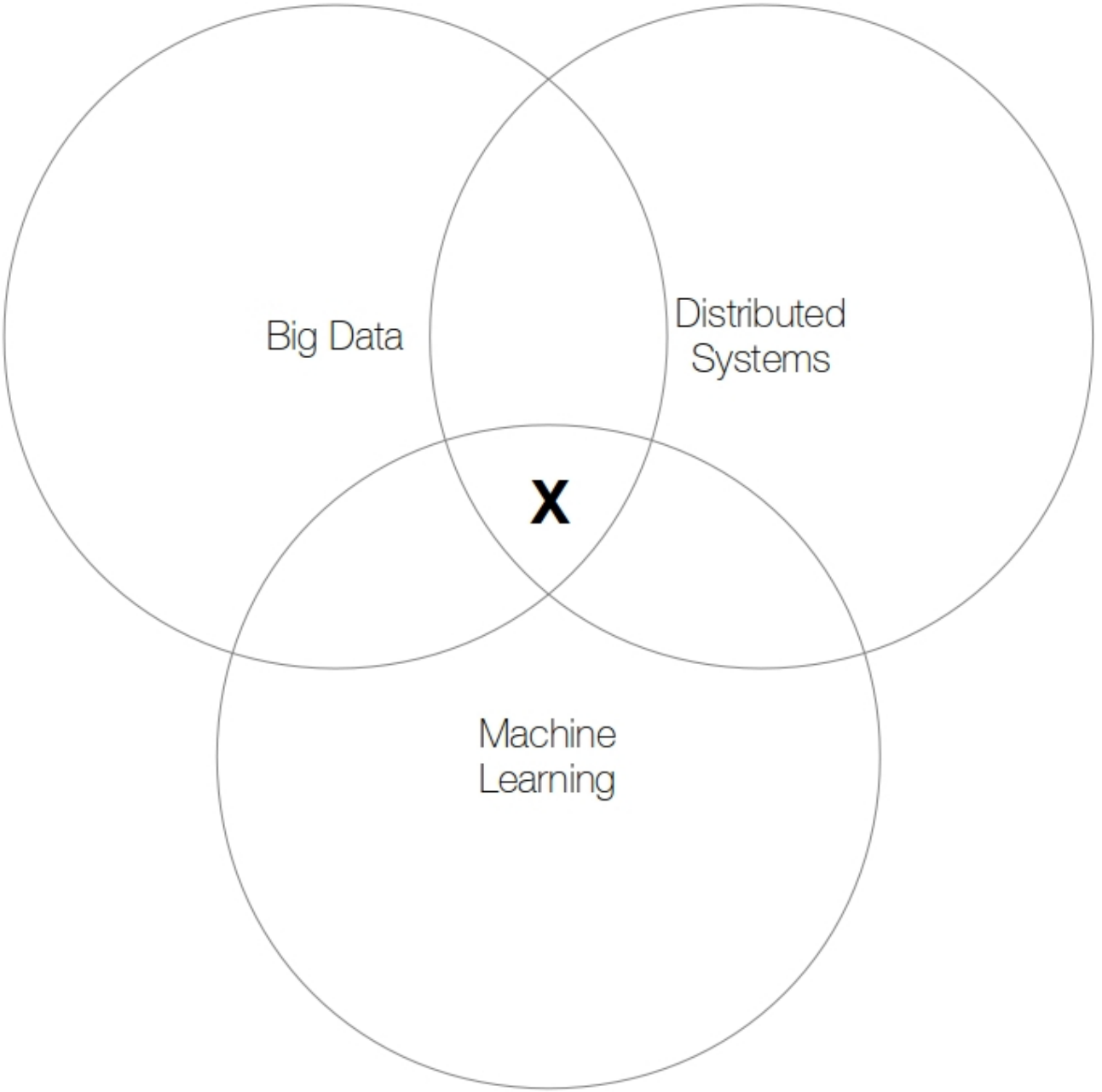
Converging Trends

Converging Trends

Converging Trends



Converging Trends



Motivation and Contribution

Why Scale Up to the Cloud?

Input data size

- Large training data instances
  - e.g DryadLINQ and MapReduce
- High input dimensionality
  - e.g GraphLab and GraphChi

Why Scale Up to the Cloud?

Complexity of data and computation

- Data complexity brings algorithm comlexity
  - e.g. PLANET (on top of MapReduce)

Why Scale Up to the Cloud?

Time constraint and parameter tuning

- Distribute system usage to increase throughput
- Model and hyper-parameter selection are repetitive and independent

Why Scale Up to the Cloud?

Data Parallelism

- MapReduce, GraphLab

Task Parallelism

- Multicores, GPUs (CUDA), MPI

or perhaps Hybrid?

- *Spark, GraphLab, DryadLINQ*

Problem?

With the various option of distributed architectures, implementing different machine systems become very task-specific

Different architectures brings different benefits and constraints

Spark + MLbase

Unified scalable machine learning

Suitable for many common Machine Learning problem

(project hypothesis)

Project Goal

Develop Mainstream ML

Evaluate Spark+MLbase on developing common Machine Learning problems

Develop Mainstream ML

Classification

- e.g. Bayesian classifier for Spam Filtering

Clustering

- e.g. k-means clustering for market segmentation

Regression

- e.g. Linear regression on weather forecasting

Collaborative Filtering

- Alternating Least Square for recommendation systems

Project Goal

Platform

- Amazon EC2

Run time

- Spark

Application

- MLbase

Project Evaluation

Evaluation and Analysis

Parallelism

- Granularity of data parallelism and task parallelism

Algorithm complexity

- Complexity of customization

Programming paradigm

- Learning curve, expressiveness and dataflow

Dataset distribution

- Management of large dataset

Performance Comparison

Learn a most suitable algorithm to be come a benchmark

Compare performance with [e.g.] GraphLab

- Speed (sequential runs)
- Scalability (efficiency increase with parallelism)
- ***Throughput (time / input size)***

Thank you!
Questions are very welcome

Haikal Pribadi
hp356@cam.ac.uk