Fuzzy Models for Data Science and Big Data

Francisco Herrera and Alberto Fernández

Soft Computing and Information Intelligent Systems (SCI²S)

University of Granada, Spain

Email: {herrera, alberto}@decsai.ugr.es

http://sci2s.ugr.es

Isaac Triguero

School of Computer Science

Automated Scheduling Optimisation and Planning (ASAP)

University of Nottingham

United Kingdom

Isaac.Triguero@nottingham.ac.uk

UNIVERSIDAD DE GRANADA

University of Nottingham

UK | CHINA | MALAYSIA

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Big Data

Our world revolves around the data

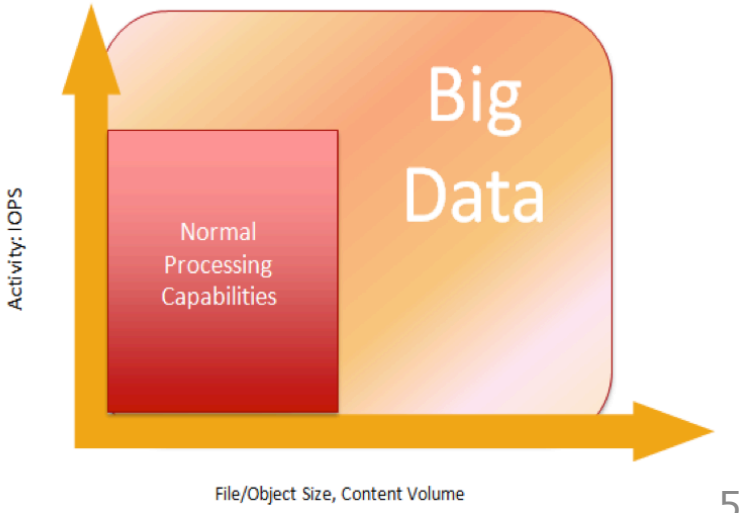
- Science
  - Data bases from astronomy, genomics, environmental data, transportation data, …
- Humanities and Social Sciences
  - Scanned books, historical documents, social interactions data, …
- Business & Commerce
  - Corporate sales, stock market transactions, census, airline traffic, …
- Entertainment
  - Internet images, Hollywood movies, MP3 files, …
- Medicine
  - MRI & CT scans, patient records, …
- Industry, Energy, …
  - Sensors, …

What is Big Data?

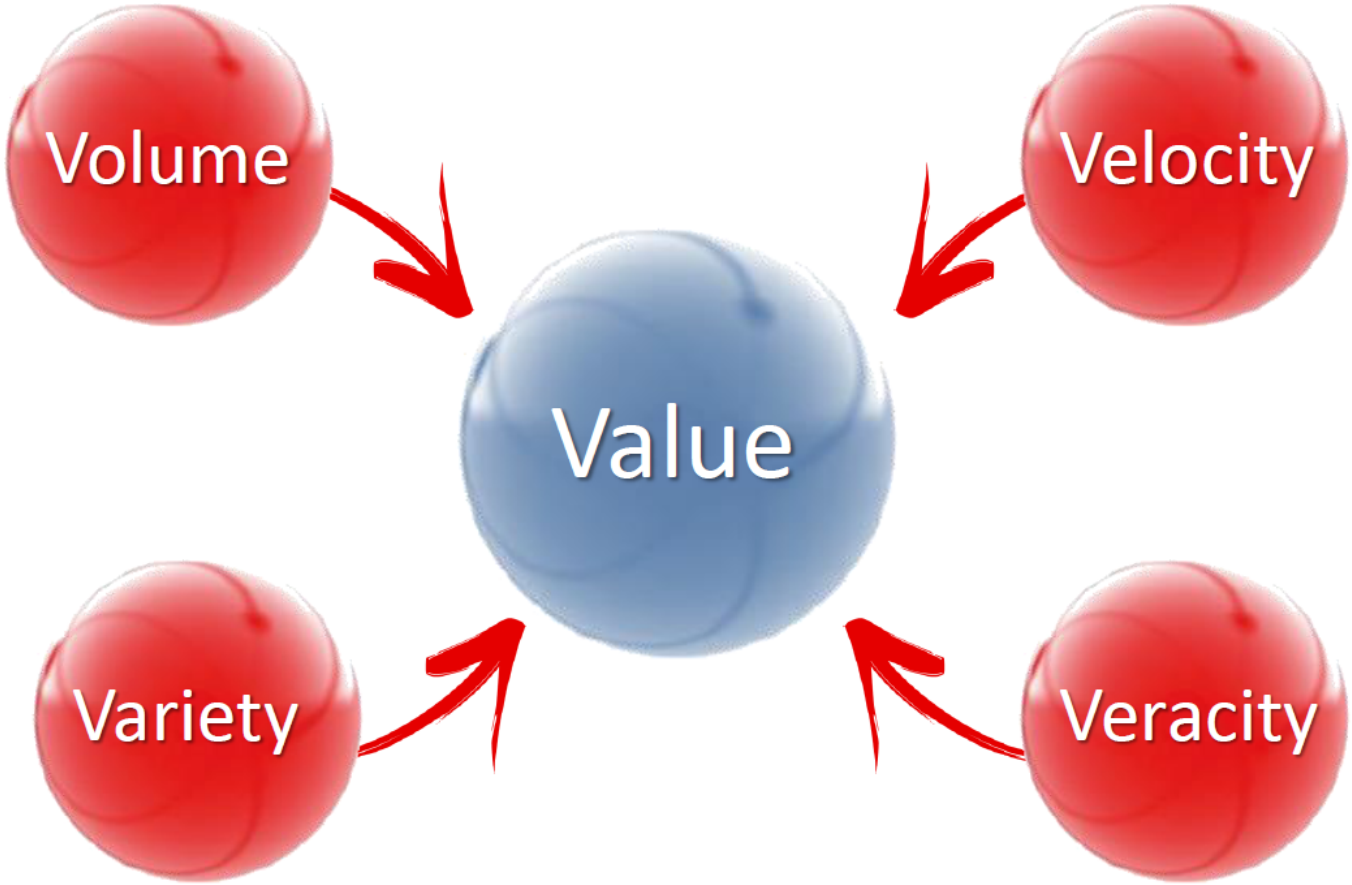
There is not a standard definition!

"*Big Data*" involves data whose volume, diversity and complexity requires new techniques, algorithms and analyses to extract valuable knowledge (hidden) .

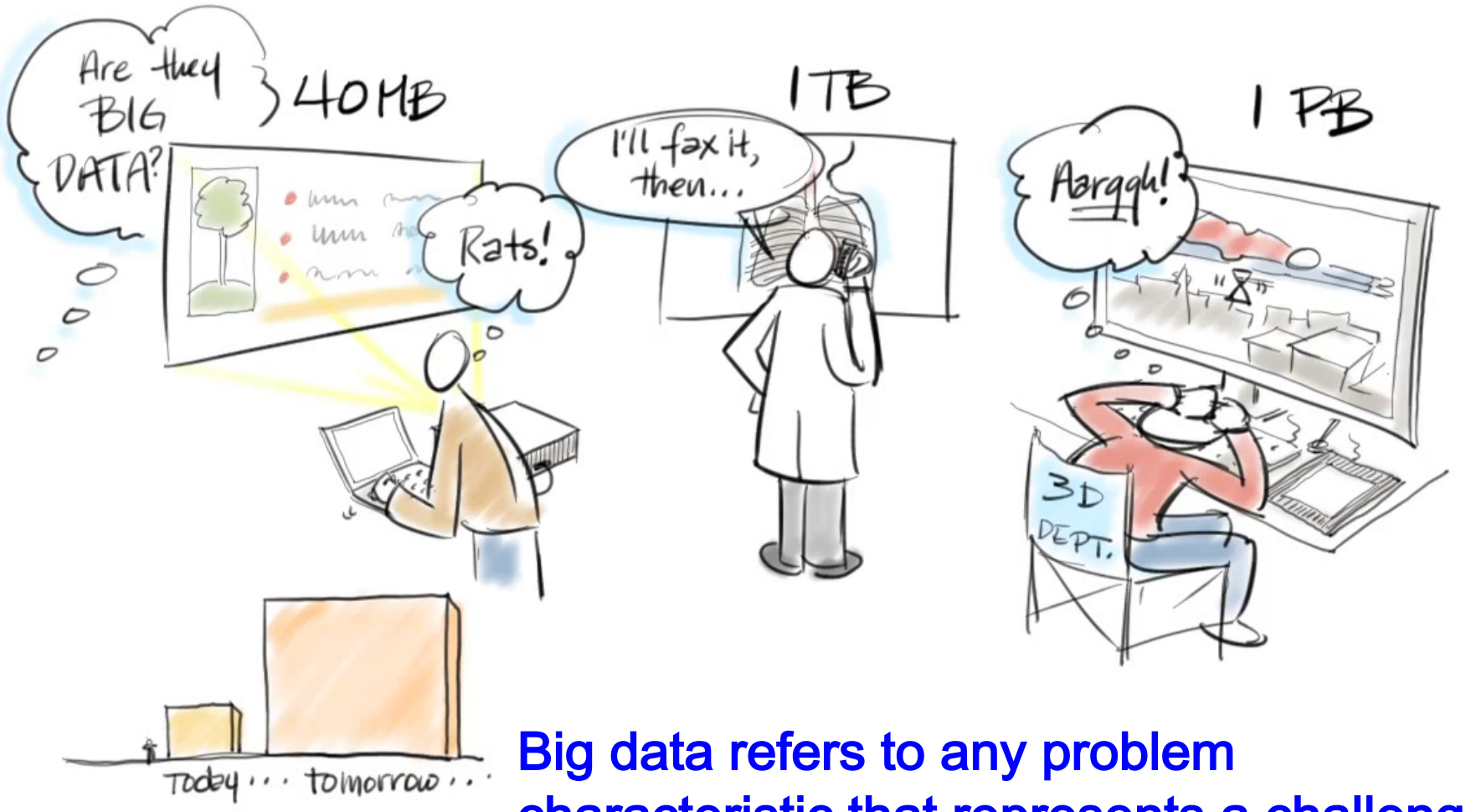
Data Intensive applications



What is Big Data? The 5V's definition

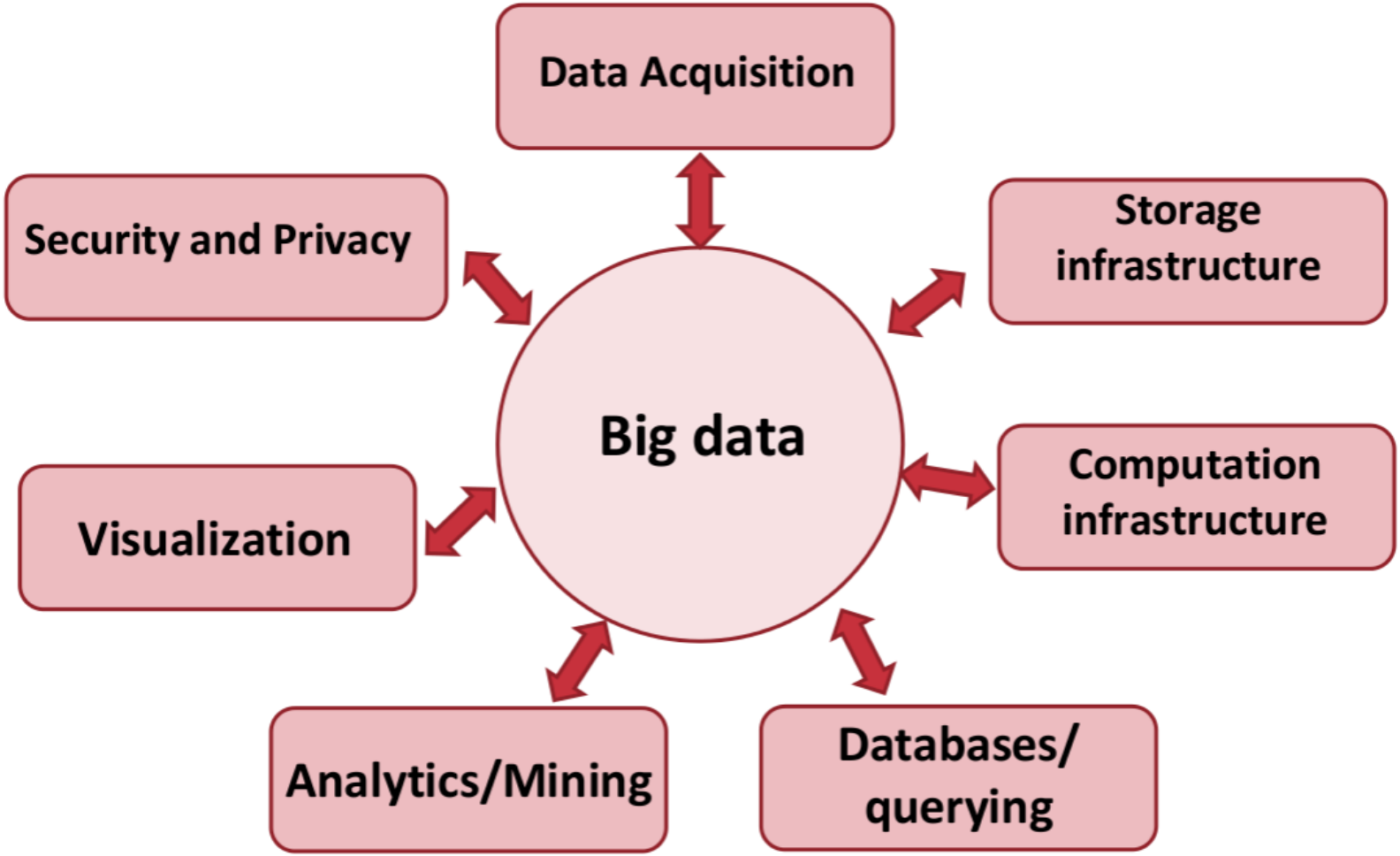


What is Big Data? (in short)

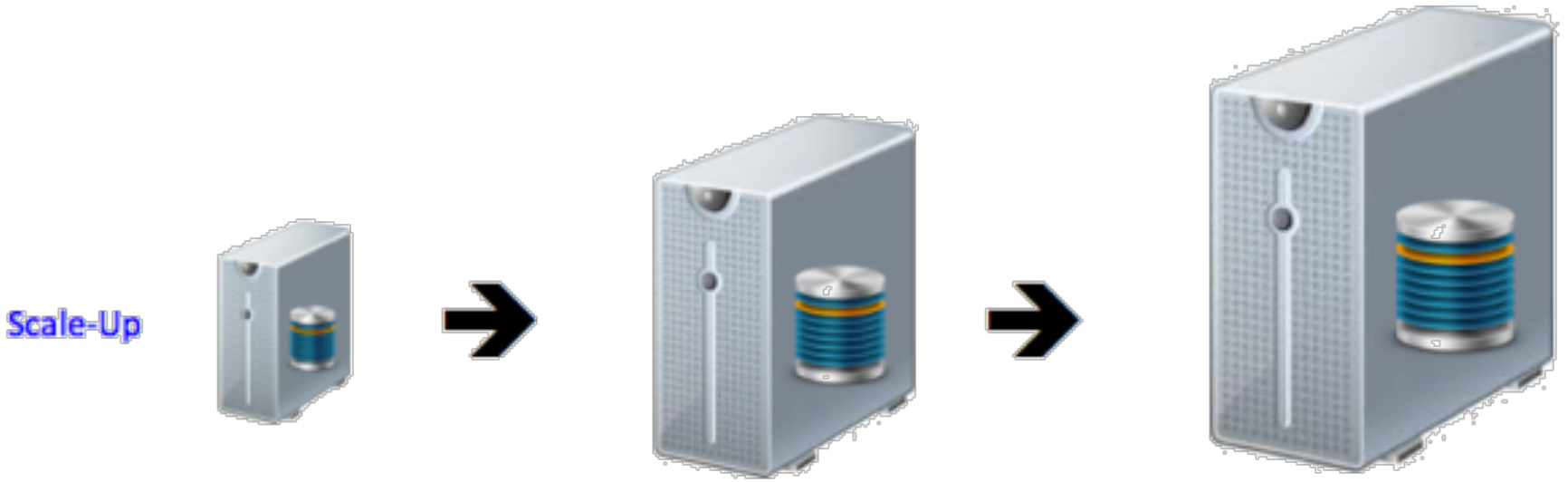


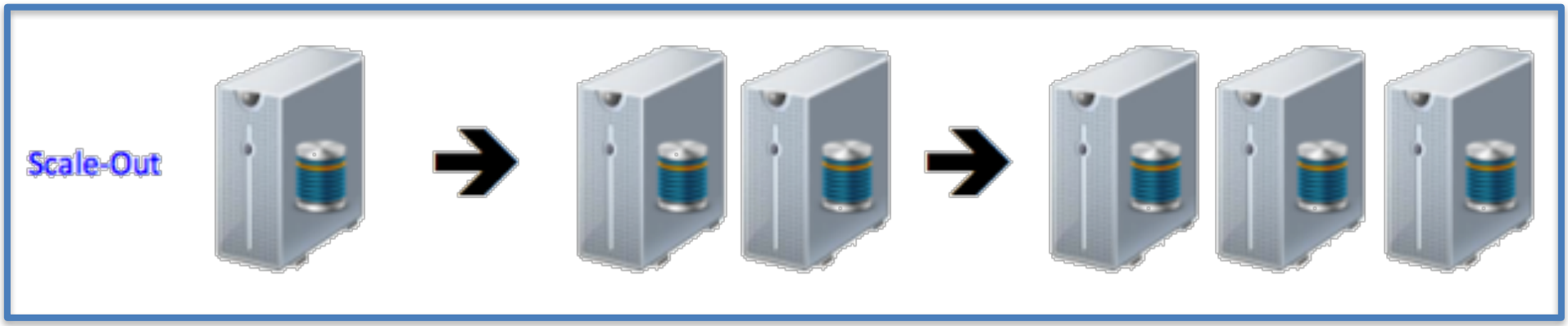
Big data refers to any problem characteristic that represents a challenge to process it with traditional applications

Big data has many faces

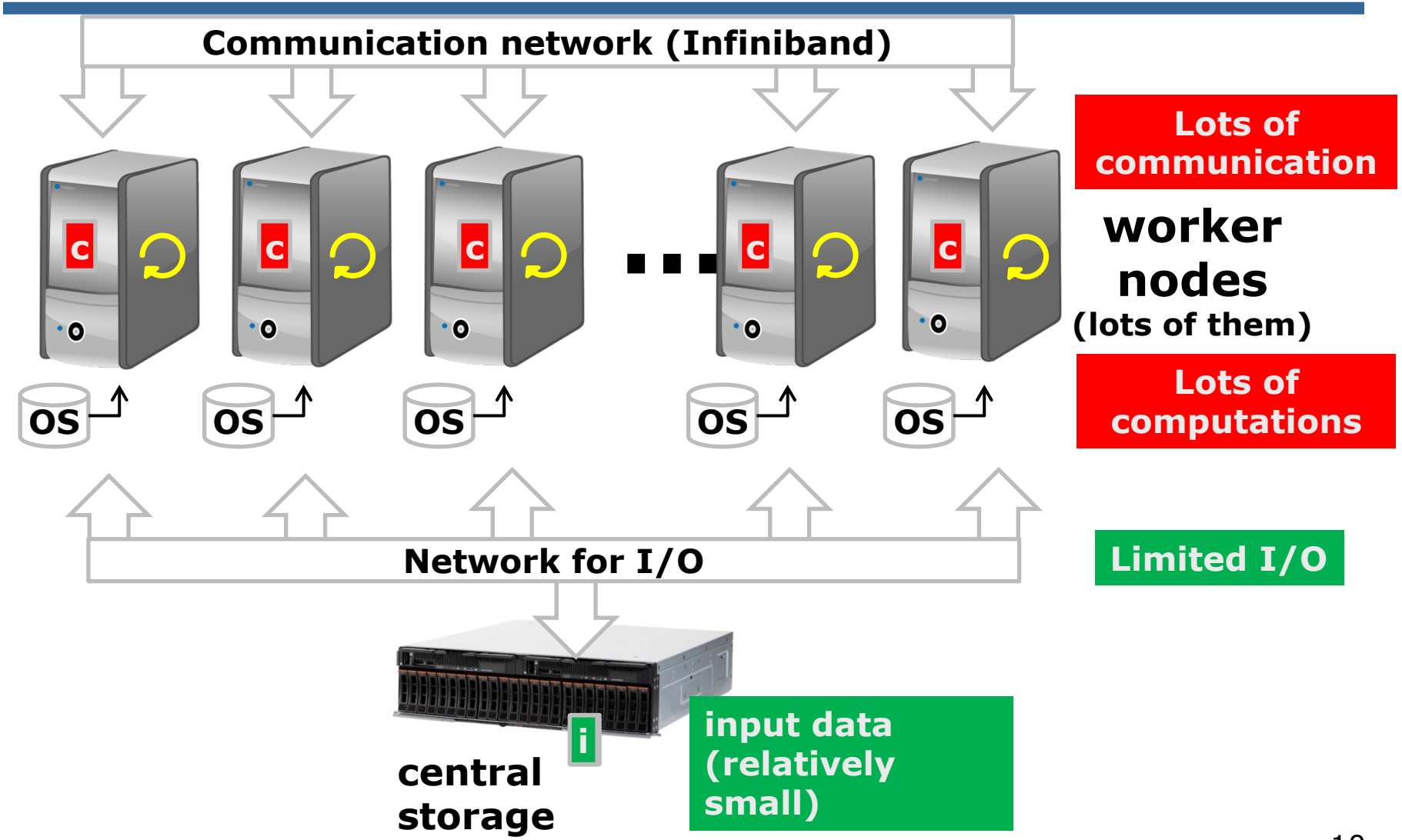


How to deal with data intensive applications? **Scale-up vs. Scale-out**



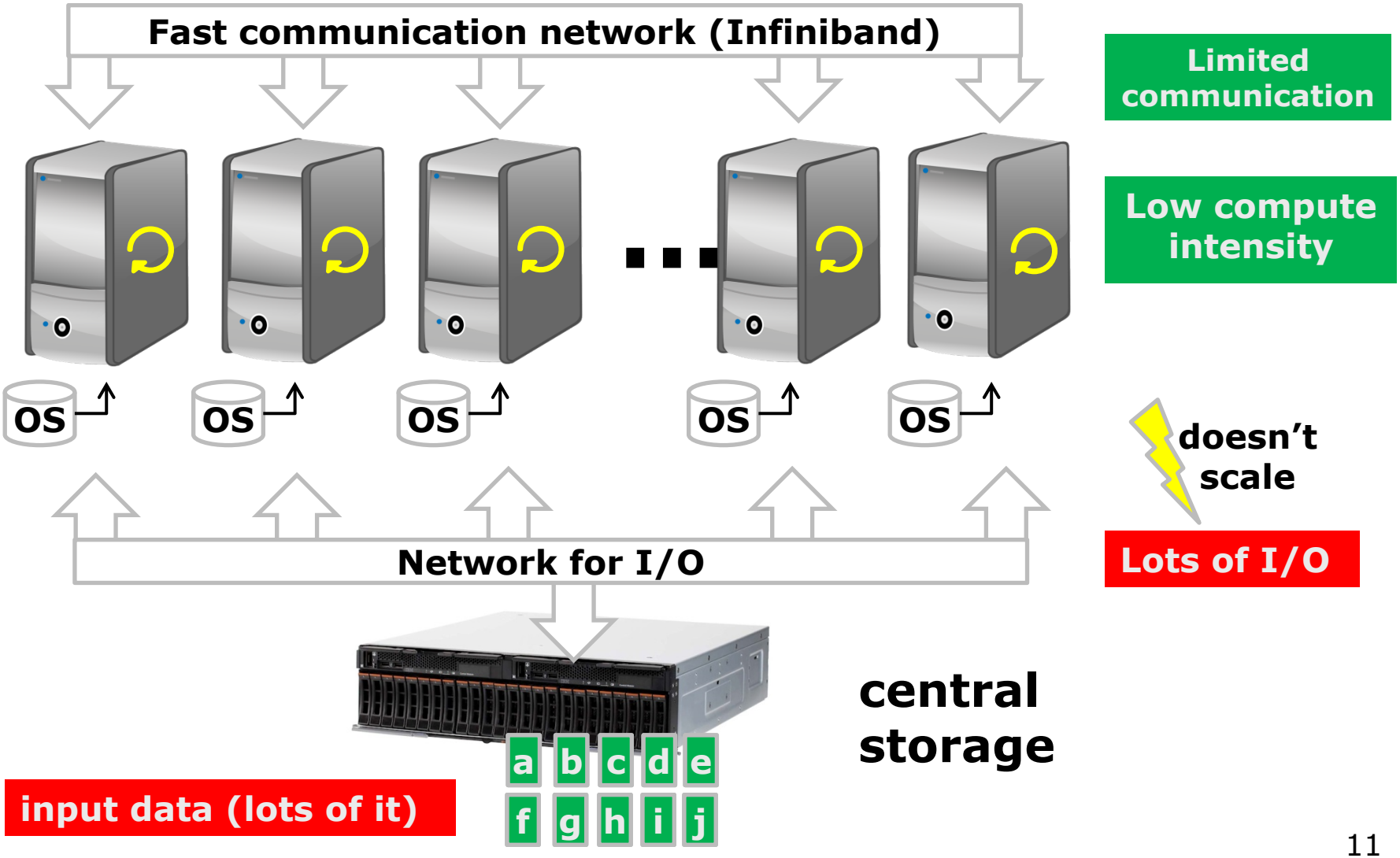


Traditional HPC way of doing things

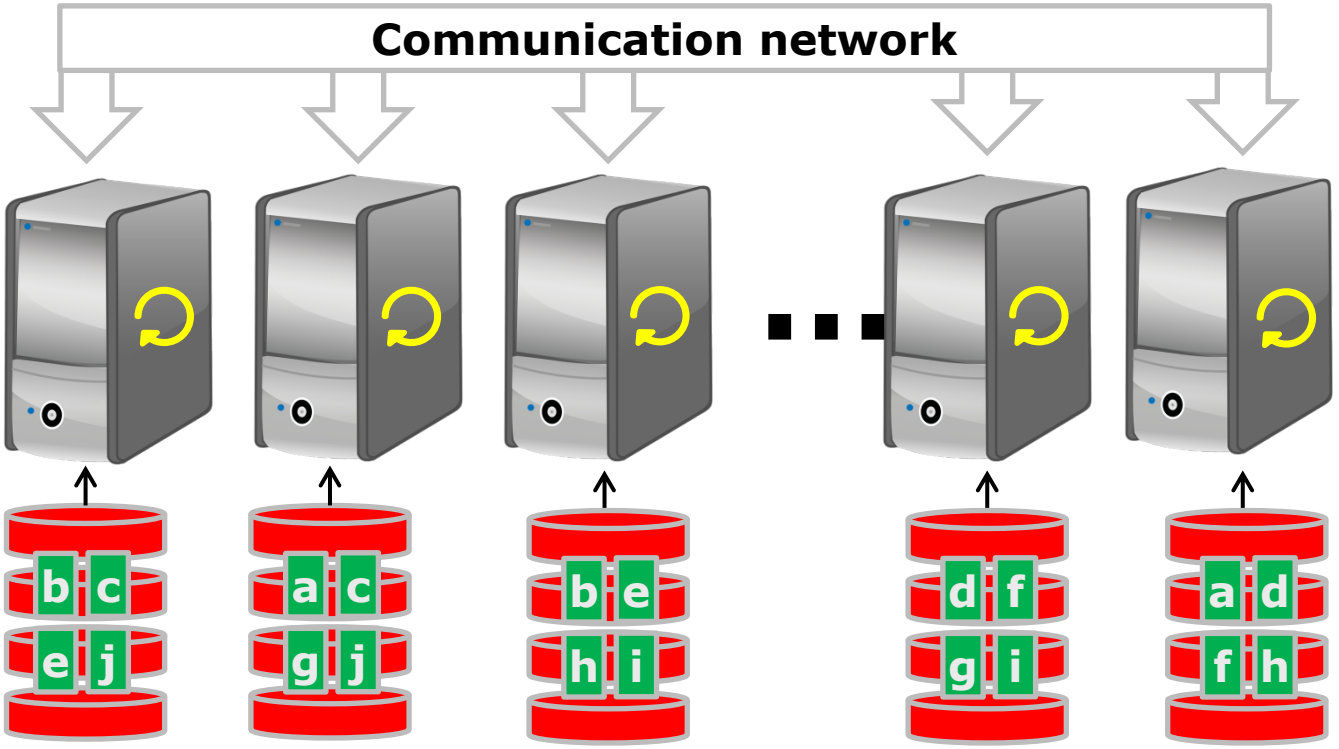


Source: Jan Fostier. Introduction to MapReduce and its Application to Post-Sequencing Analysis

Data-intensive jobs



Data-intensive jobs



Limited communication

Low compute intensity

input data (lots of it)

Solution: store data on local disks of the nodes that perform computations on that data ("data locality")

Distributed systems in Big Data

- **Objective: To apply an operation to all data**
  - One machine cannot process or store all data
    - Data is distributed in a cluster of computing nodes
    - It does not matter which machine executes the operation
    - It does not matter if it is run twice in different nodes (due to failures or straggler nodes)
    - We look for an abstraction of the complexity behind distributed systems
  - **DATA LOCALITY is crucial**
    - Avoid data transfers between machines as much as possible

Distributed systems in Big Data

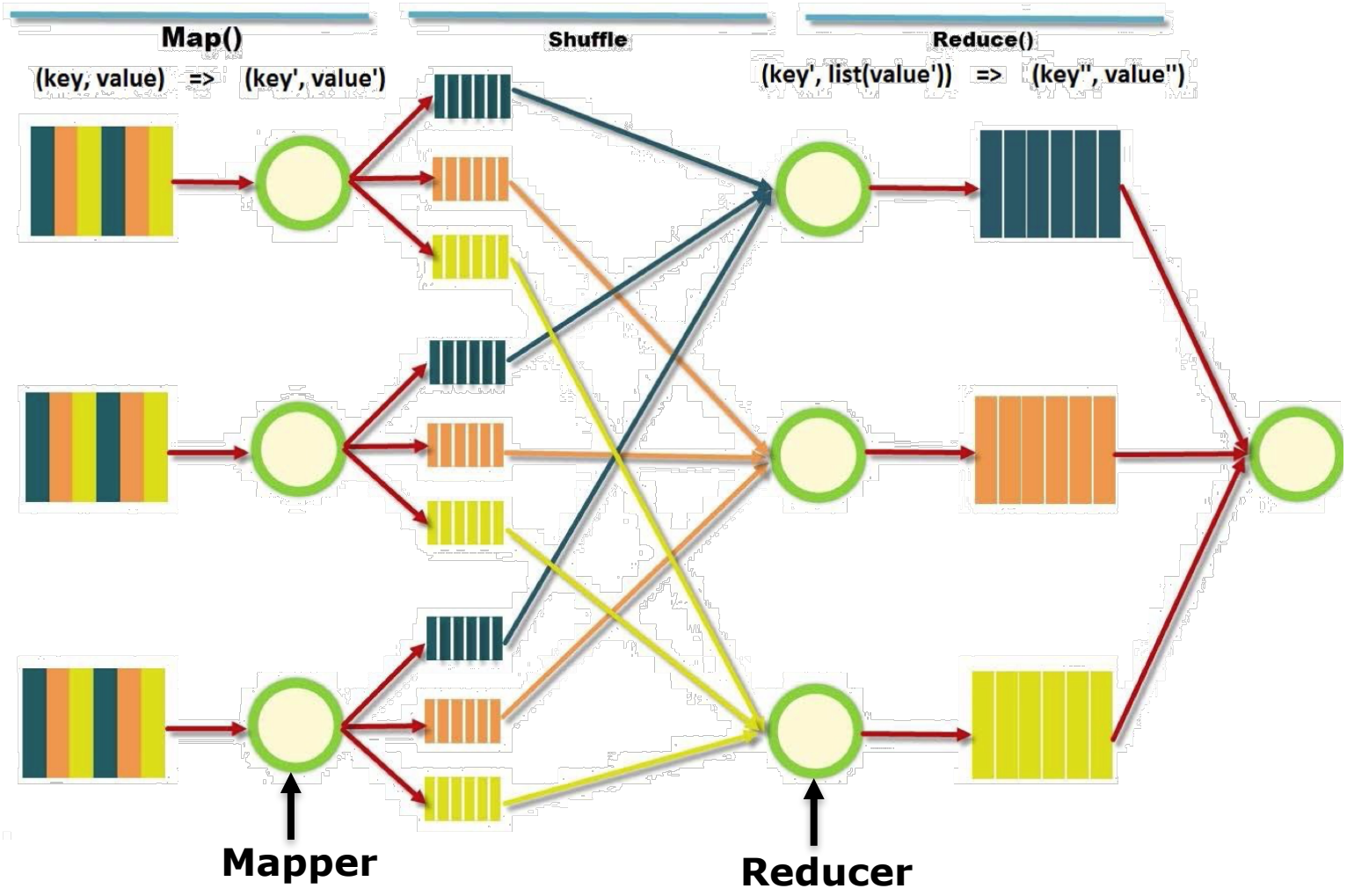
New programming model: **MapReduce**

- *"Moving computation is cheaper than moving computation and data at the same time"*
- **Idea**
  - Data is distributed among nodes (distributed file system)
  - Functions/operations to process data are distributed to all the computing nodes
  - Each computing node works with the data stored in it
  - Only the necessary data is moved across the network

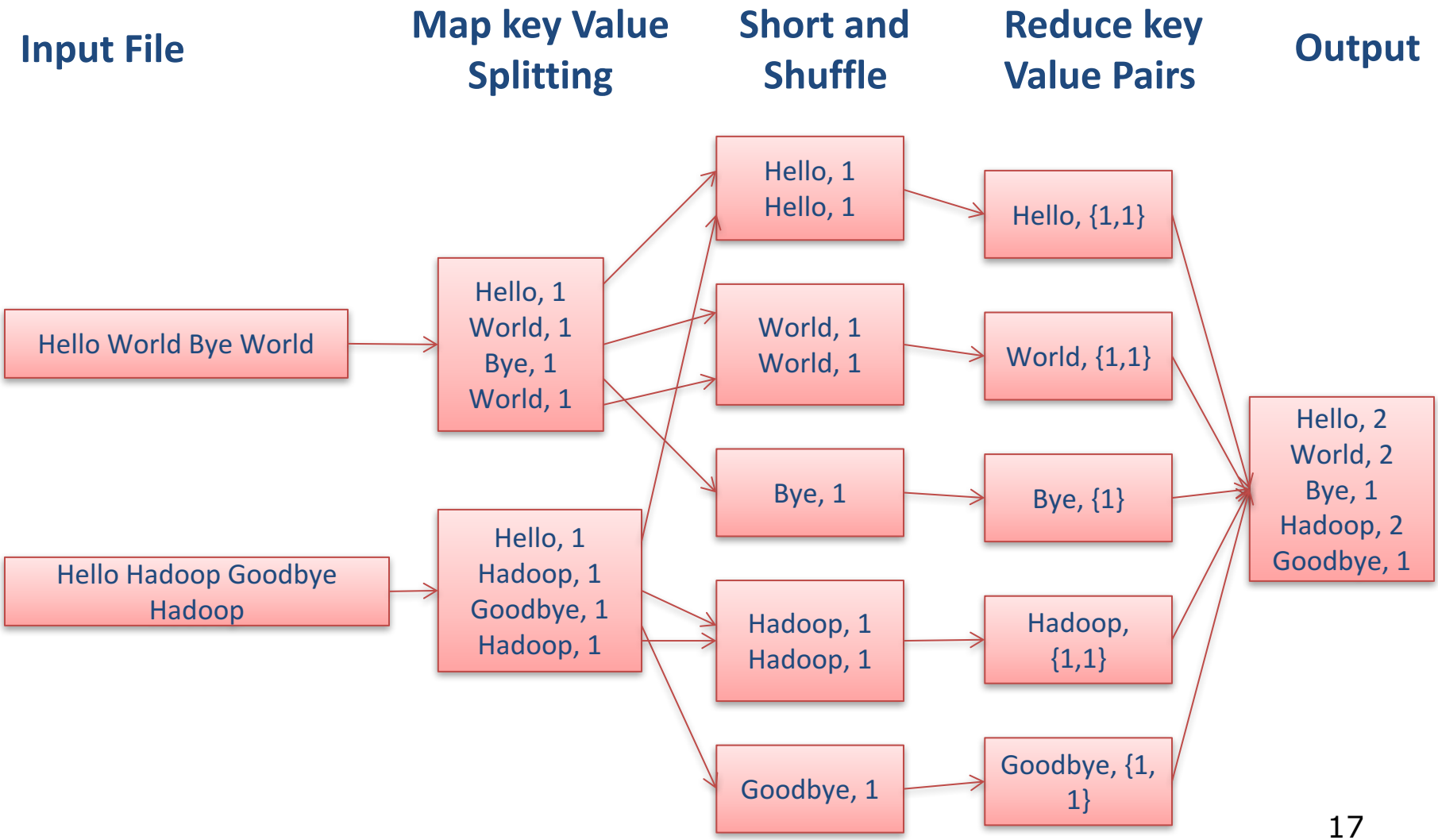
MapReduce

- Parallel Programming model
- **Divide & conquer strategy**
  - **divide**: partition dataset into smaller, independent chunks to be processed in parallel (*map*)
  - **conquer**: combine, merge or otherwise aggregate the results from the previous step (*reduce*)
- Based on **simplicity** and **transparency** to the programmers, and assumes **data locality**
- Becomes popular thanks to the open-source project Hadoop! (Used by Google, Facebook, Amazon, …)

MapReduce: How it works



MapReduce: WordCount Example



Hadoop

- ***Open source framework for Big Data processing***
  - **Based on** two Works published by Google
    - **Google File System** (GFS)[Ghe03]
    - **MapReduce** algorithm[Dea04]
  - **Composed of**
    - **Hadoop Distributed File System** (HDFS) ➔ Storage
    - **Implementation of the MapReduce algorithm** ➔ Processing

[Ghe03] S. Ghemawat, H. Gobioff, S.-T. Leung. The Google file system. In Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03). ACM, New York, NY, USA, 29-43. 2003
[Dea04] J. Dean, S. Ghemawat. MapReduce: simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6 (OSDI'04), Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10. 2004.

Hadoop

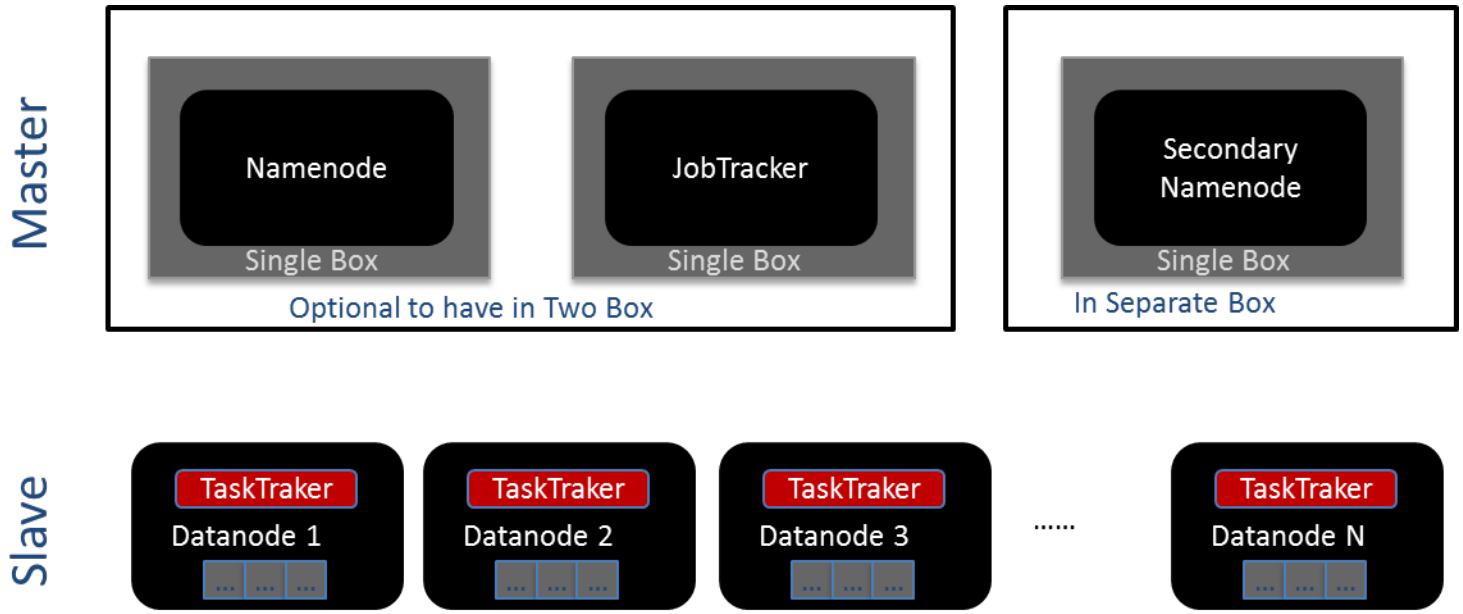
- Hadoop is:
  - An **open-source** framework written in Java
  - Distributed storage of very large data sets (Big Data)
  - Distributed processing of very large data sets

- This framework consists of a **number of modules**
  - *Hadoop Common*
  - ***Hadoop Distributed File System (HDFS)***
  - *Hadoop YARN* – resource manager
  - ***Hadoop MapReduce*** – programming model

http://hadoop.apache.org/

Hadoop: A master/slave architecture

- **Master**: NameNode, JobTracker
- **Slave**: {DataNode, TaskTraker}, ..., {DataNode, TaskTraker}



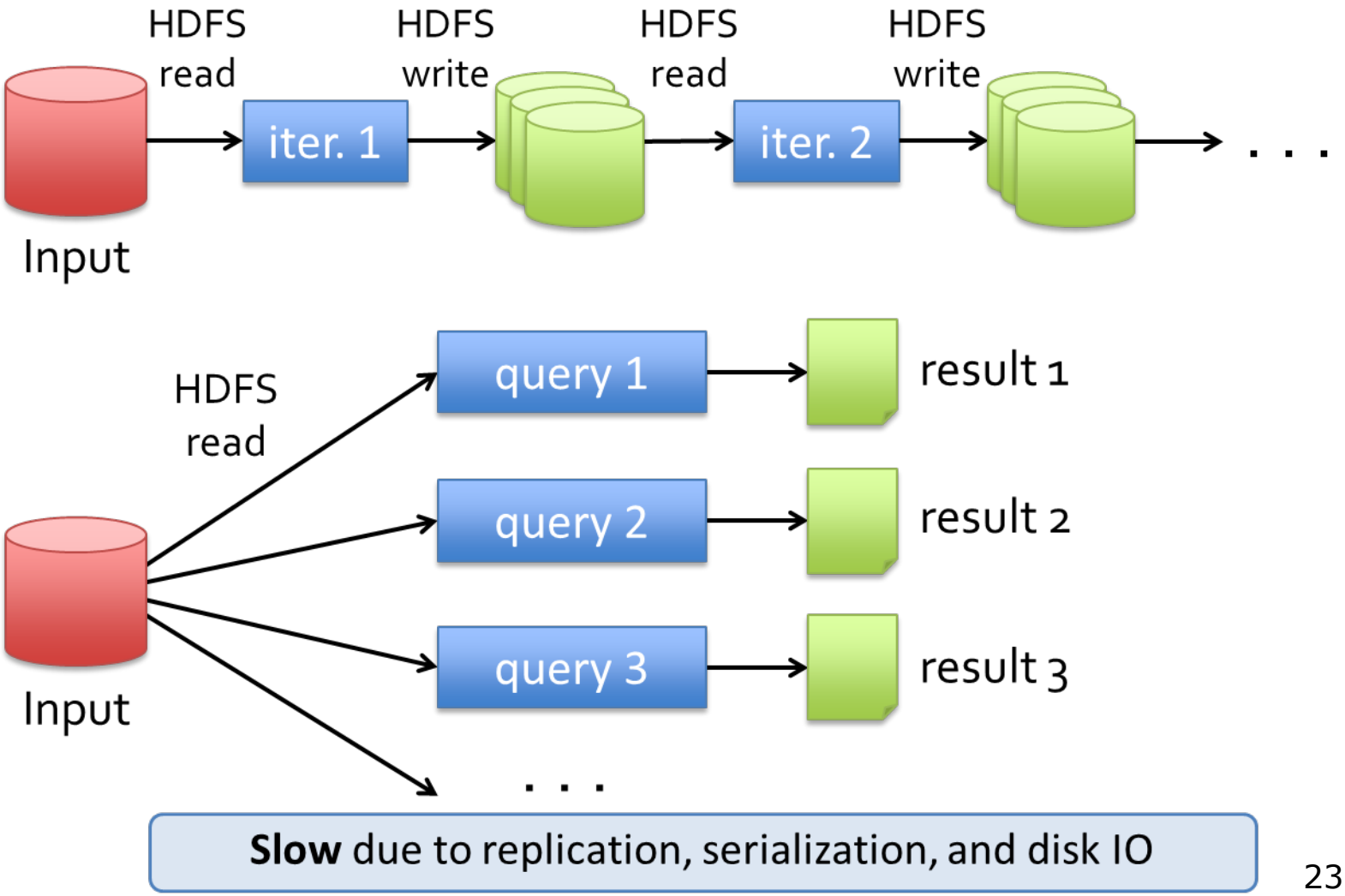
Distributed File System: HDFS

- **HDFS – Hadoop Distributed File System**
  - Distributed File System written in Java
  - Scales to clusters with **thousands of computing nodes**
    - Each node stores part of the data in the system
  - **Fault tolerant** due to data replication
  - Designed for big files and low-cost hardware
    - GBs, TBs, PBs
  - **Efficient for read and append operations** (random updates are rare)
  - High throughput (for bulk data) more important than low latency

Hadoop MapReduce: Main Characteristics

- **Automatic parallelization:**
  - Depending on the size of the input data ➔ there will be multiple MAP tasks!
  - Depending on the number of Keys <k, value> ➔ there will be multiple REDUCE tasks!
- **Scalability:**
  - It may work over every data center or cluster of computers.
- **Transparent for the programmer**
  - Fault-tolerant mechanism.
  - Automatic communications among computers

Data Sharing in Hadoop MapReduce

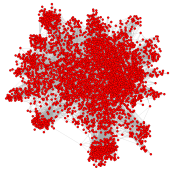


Paradigms that do not fit with Hadoop MapReduce

- **Directed Acyclic Graph (DAG) model:**
  - The DAG defines the dataflow of the application, and the vertices of the graph defines the operations on the data
- **Graph model:**
  - More complex graph models that better represent the dataflow of the application
  - Cyclic models -> Iterativity.
- **Iterative MapReduce model:**
  - An extented programming model that supports iterative MapReduce computations efficiently

New platforms to overcome Hadoop's limitations

GIRAPH (APACHE Project)
(http://giraph.apache.org/)
Iterative graph processing



GPS - A Graph Processing System, (Stanford)
http://infolab.stanford.edu/gps/
Amazon's EC2

Distributed GraphLab (Carnegie Mellon Univ.)
https://github.com/graphlab-code/graphlab
Amazon's EC2

Spark (UC Berkeley)
(100 times more efficient than Hadoop, including iterative algorithms, according to creators)
http://spark.incubator.apache.org/research.html

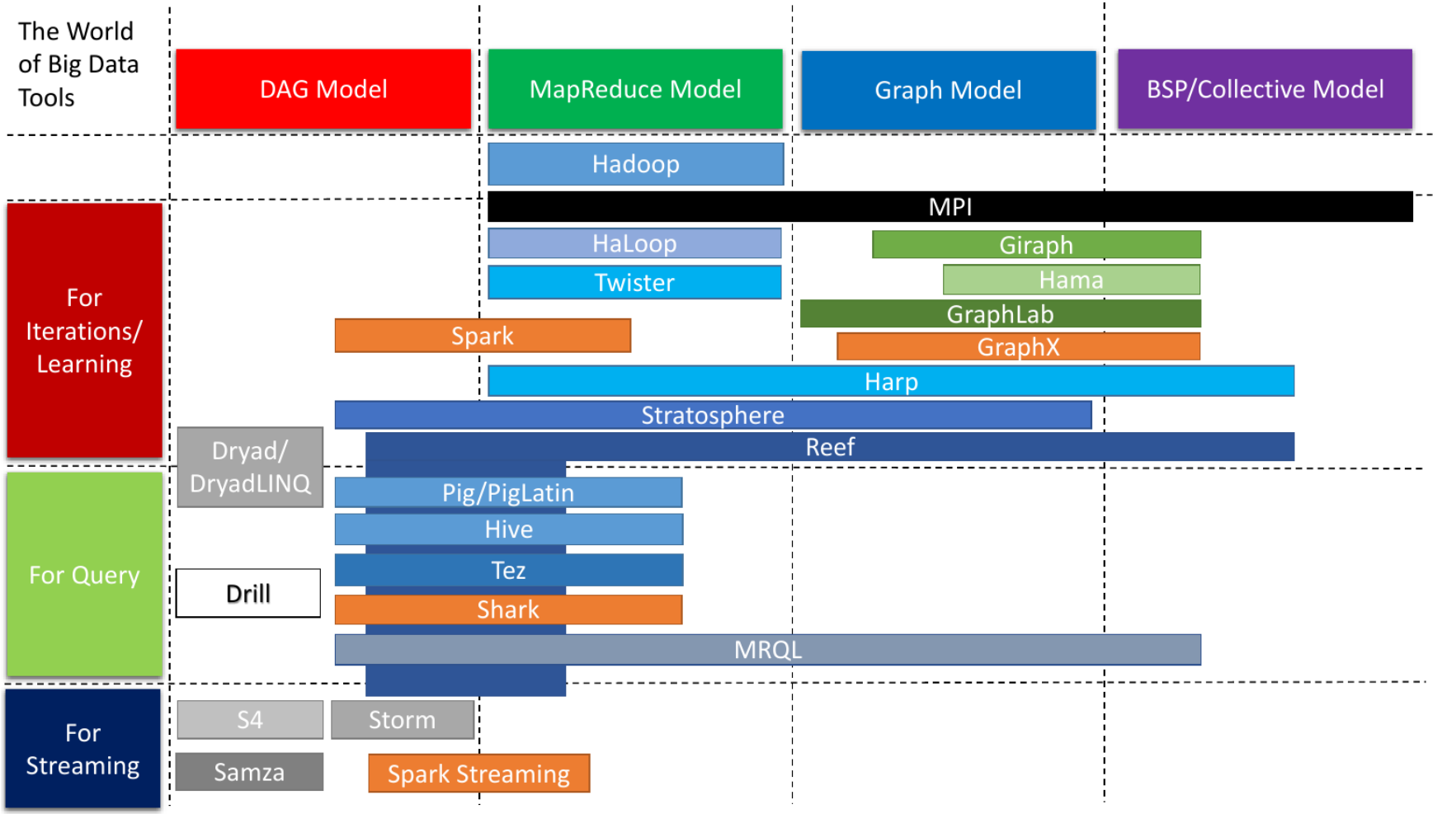
Twister (Indiana University)
http://www.iterativemapreduce.org/
Private Clusters

PrIter (University of Massachusetts Amherst, Northeastern University-China)
http://code.google.com/p/priter/
Private cluster and Amazon EC2 cloud

HaLoop (University of Washington)
http://clue.cs.washington.edu/node/14
http://code.google.com/p/haloop/
Amazon's EC2

GPU based platforms
Mars
Grex

Big data technologies



What is Spark?

Fast and Expressive Cluster Computing Engine Compatible with Apache Hadoop

Up to **10×** faster on disk,
100× in memory

2-5× less code

Efficient

- General execution graphs
- In-memory storage

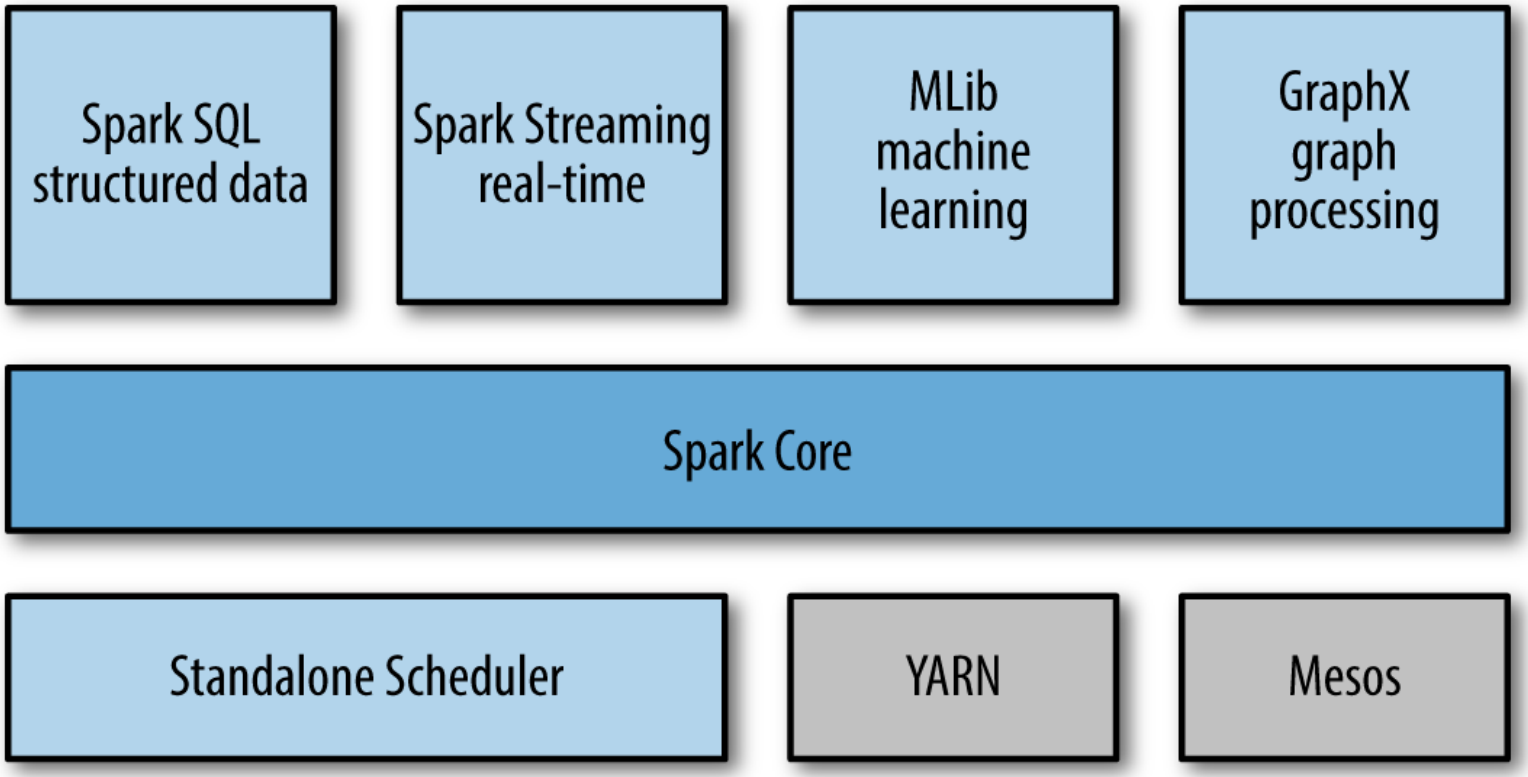
Usable

- Rich APIs in Java, Scala, Python
- Interactive shell

What is Spark?

- Data processing engine (only)
- Without a distributed file system
  - Uses other existing DFS
    - HDFS, NoSQL…
    - Hadoop is not a prerequisite
- Works with different cluster management tools
  - Hadoop (YARN)
  - Mesos
  - Standalone mode (included in Spark)

What is Spark?



Spark Goal

- Provide distributed memory abstractions for clusters to support apps with working sets
- **Retain the attractive properties of MapReduce**:
  - Fault tolerance (for crashes & stragglers)
  - Data locality
  - Scalability

Solution: augment data flow model with "resilient distributed datasets" (RDDs)

RDDs in Detail

- An RDD is a fault-tolerant collection of elements that can be operated on in parallel.
- There are two ways to create RDDs:
  - Parallelizing an existing collection in your driver program
  - Referencing a dataset in an external storage system, such as a shared filesystem, HDFS, Hbase.
- *Can be cached for future reuse*

Operations with RDDs

- Transformations (e.g. map, filter, groupBy, join)
  - Lazy operations to build RDDs from other RDDs
- Actions (e.g. count, collect, save)
  - Return a result or write it to storage

Transformations (define a new RDD)
map filter sample union groupByKey reduceByKey join cache …

Parallel operations (return a result to driver)
reduce collect count save lookupKey …

Operations with RDDs

- **RDDs – Simple example**

```
>>> lines = sc.textFile("README.md") # Creates an RDD
>>> lines.count() # Counts the number of elements in the RDD
127
>>> lines.first() # 1st element of the RDD -> 1st line of
README.md
u'# Apache Spark'
```

- Simple wordCount in Spark

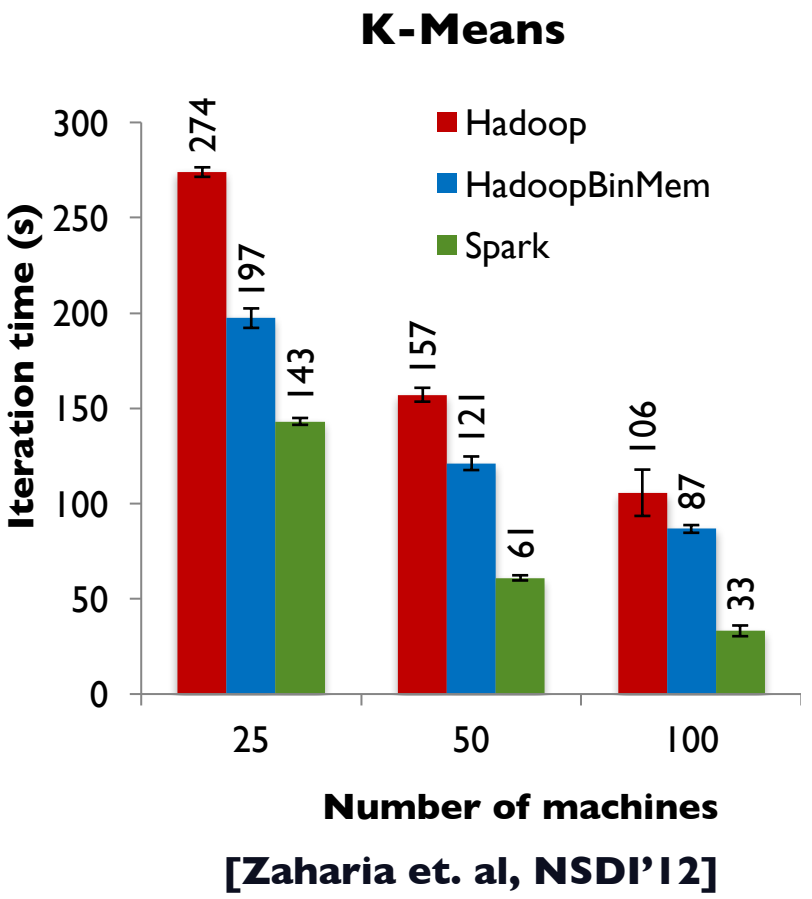
```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" "))
             .map(lambda word: (word, 1))
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

Spark vs. hadoop

Lines of code for K-Means

Spark ~ 90 lines –

Hadoop ~ 4 files, > 300 lines

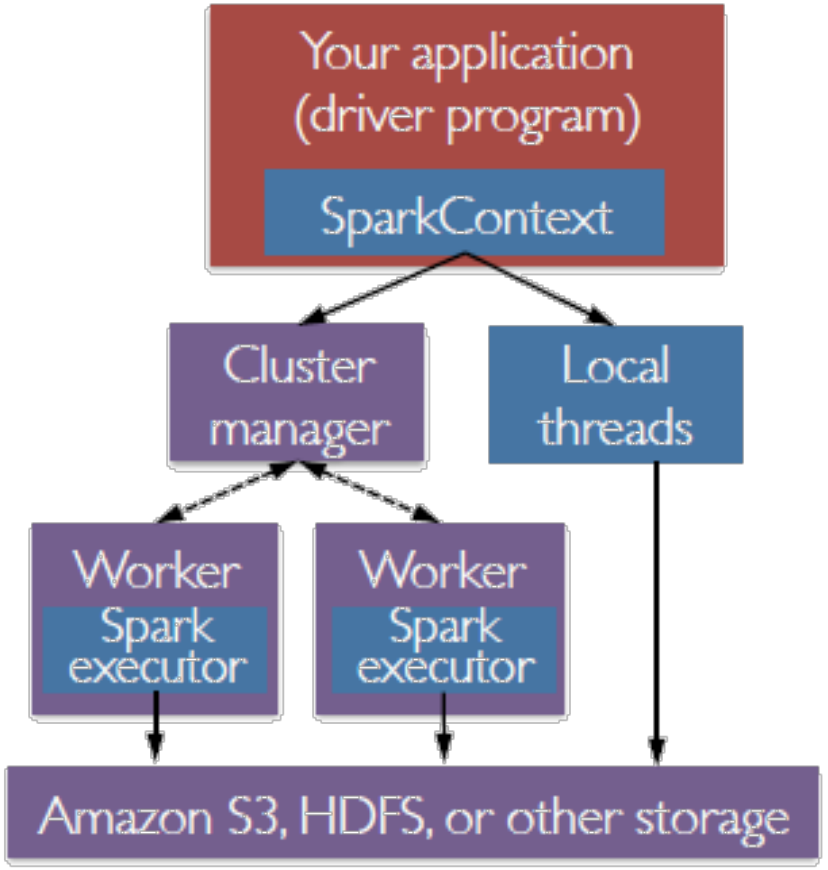


[Zaharia et. al, NSDI'12]

M. Zaharia et al. **Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing**. NSDI 2012.

Spark

- **Driver** and **Workers**
  - A **Spark program is composed of two programs**
    - **Driver program**
    - **Workers program**
      - Executed in the computing nodes
      - Or in local threads
  - RDDs are distributed across the whole cluster



SparkSQL: Datasets & DataFrames

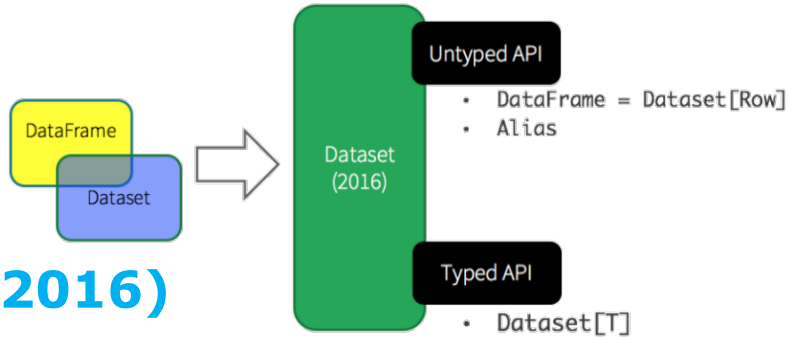
- **Structured APIs**
  - **DataFrames (>= Spark 1.3)**
    - Idea: **RDDs of rows with columns that can be accessed by their names**
    - **Similar to Pandas in Python (dataframes in R)**
    - Avoid Java serialization performed by RDDs
    - API natural for developers familiar with building query plans (SQL)
    - Introduced as a part of Tungsten project
      - Efficient memory management
    - Concept of schema to describe data

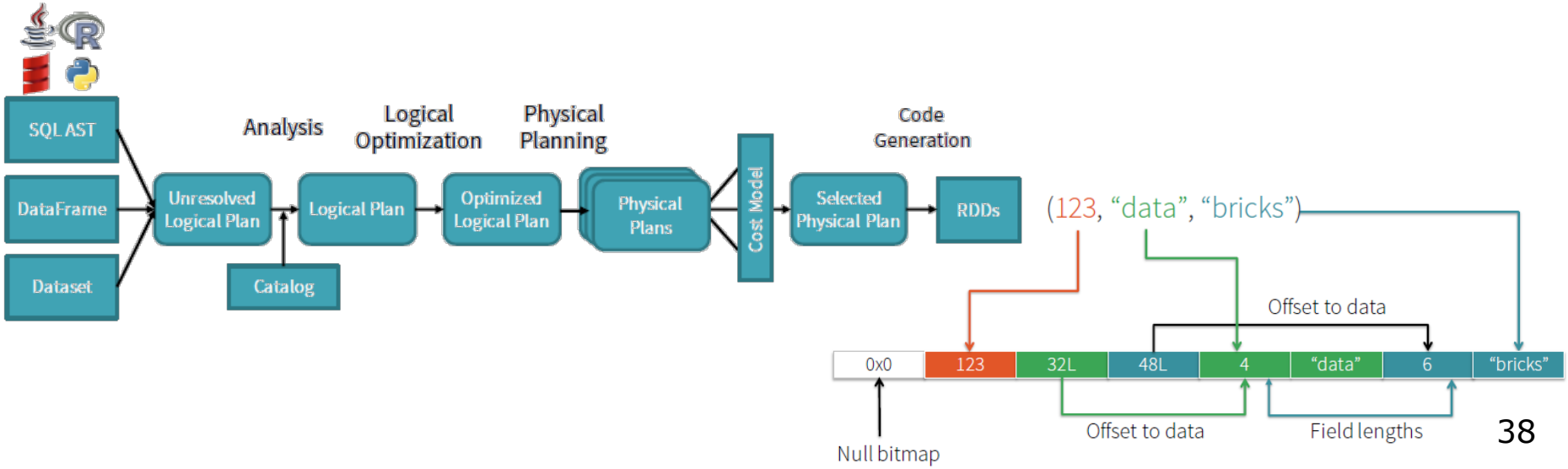
SparkSQL: Datasets & DataFrames

- **Structured APIs**
  - **Datasets (>= Spark 1.6)**
    - Idea: Strongly typed RDDs
    - **Functional transformations (map, flatMap, filter)**
    - **Best of both RDDs and DataFrames**
      - Object-oriented programming
      - Compile-time type safety
      - Catalyst optimization
      - Tungsten off-heap memory optimization
    - Only for Scala and Java

SparkSQL: Datasets y DataFrames

- **Structured APIs**
  - **DataFrames and Datasets**
  - **Fused in Spark 2.0 (November 2016)**
    - A DataFrame is just a Dataset of Rows: Dataset[Row]
  - Both make use of Catalyst and Tungsten projects





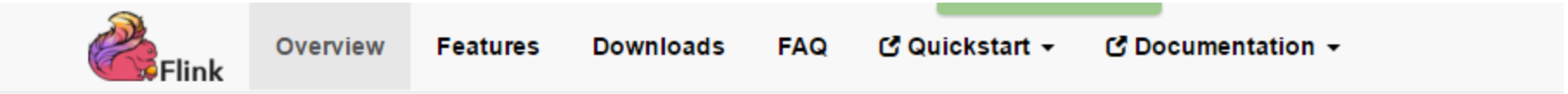
SparkSQL: Datasets y DataFrames

- **Structured APIs in Spark**
  - Analysis of the **reported errors** before a job is executed

	SQL	DataFrames	Datasets
Syntax Errors	Runtime	Compile Time	Compile Time
Analysis Errors	Runtime	Runtime	Compile Time

Flink

https://flink.apache.org/



Apache Flink is an open source platform for distributed stream and batch data processing.

Flink's core is a streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams.

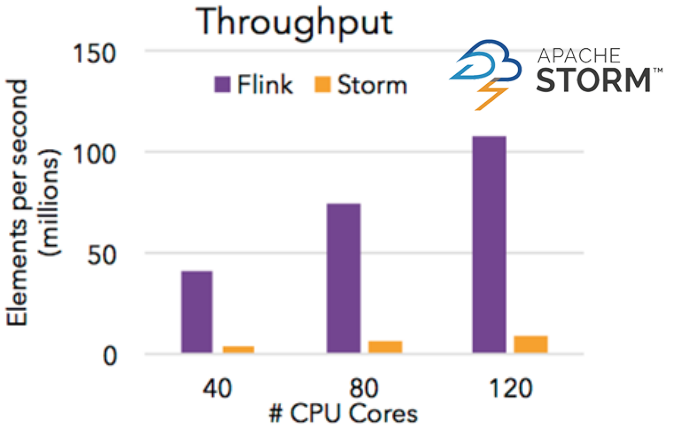
Flink includes **several APIs** for creating applications that use the Flink engine:

1. DataStream API for unbounded streams embedded in Java and Scala, and
2. DataSet API for static data embedded in Java, Scala, and Python,
3. Table API with a SQL-like expression language embedded in Java and Scala.

Flink also bundles **libraries for domain-specific use cases**:

1. CEP, a complex event processing library,
2. Machine Learning library, and
3. Gelly, a graph processing API and library.

You can **integrate** Flink easily with other well-known open source systems both for data input and output as well as deployment.



Batch on Streaming

Batch processing applications run efficiently as special cases of stream processing applications.

Streaming First

High throughput and low latency stream processing with exactly-once guarantees.

APIs, Libraries, and Ecosystem

DataSet, DataStream, and more. Integrated with the Apache Big Data stack.

2001-2010

2010-2017

Big Data: Technology and Chronology

The Apache Software Foundation

2001
3V's Gartner
Doug Laney

Google
MapReduce

Hadoop
Ecosystem

2009-2013 Flink
TU Berlin
Flink Apache (Dec. 2014) Volker Markl

2004
MapReduce
Google
Jeffrey Dean

Flink

Big Data

hadoop
YAHOO!

Spark

2010 Spark
U Berckeley
Apache Spark
Feb. 2014
Matei Zaharia

2008
Hadoop
Yahoo!
Doug Cutting

2010-2017:

Big Data Analytics: Mahout, MLLib, …

Hadoop Ecosystem

Applications

New Technology

Big Data Ecosystem

Cluster management

Cluster management

Pig
Scripting
para
MapReduce

Sqoop
Hadoop to RDBMS

Data serialization

SQL

NoSQL Database

Streaming data

Streaming data

DAG execution

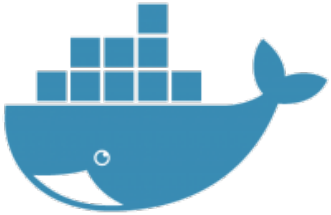
ZooKeeper
Coordinador

NoSQL on HDFS

Kafka
Gestión de
fuentes de
datos

Impala
Interactive SQL

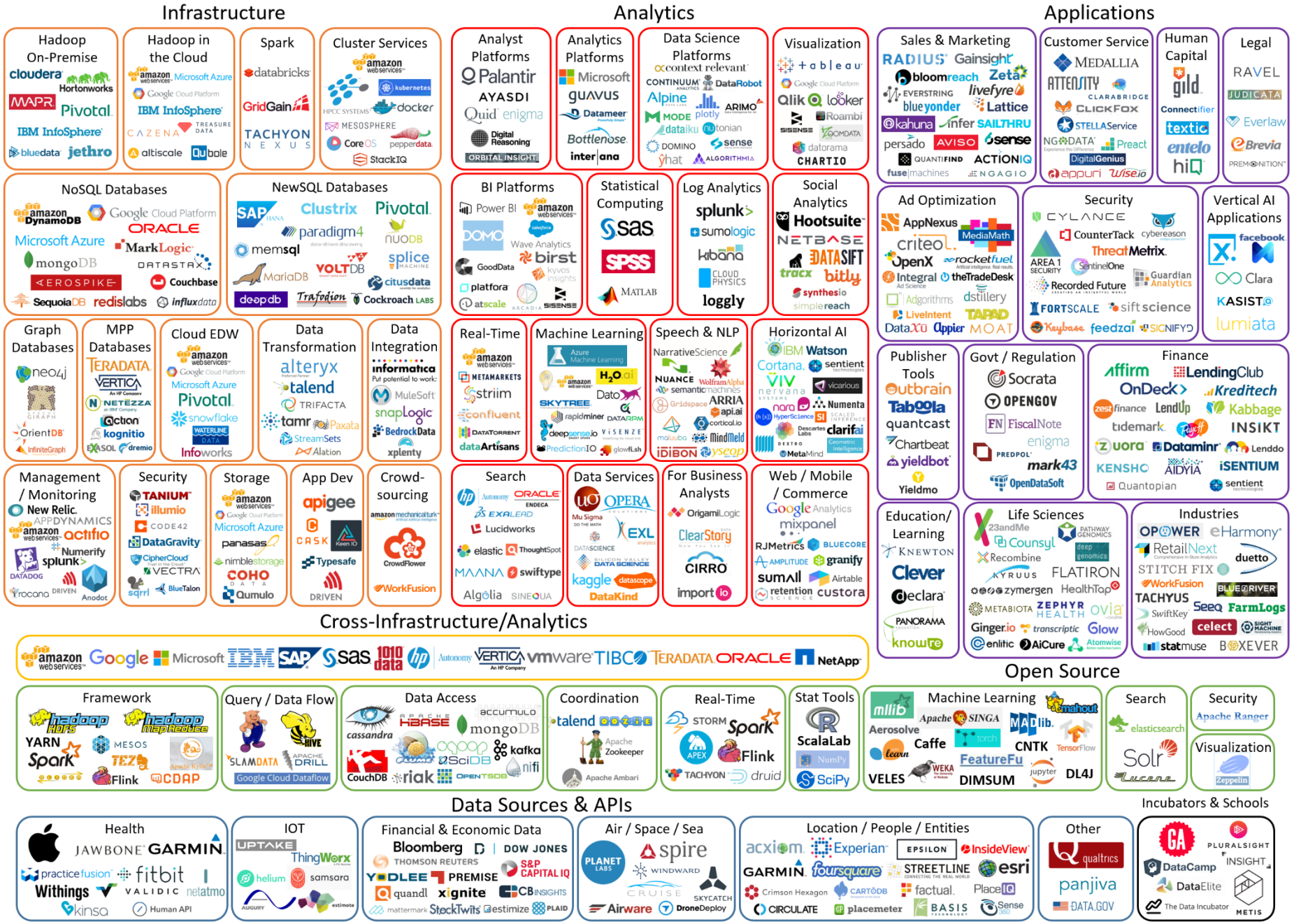
Zepelin
Web-based notebook



Docker
Containers

Log data

Big Data Landscape 2016 (Version 3.0)



Last Updated 3/23/2016

© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

FIRSTMARK

Take-home message so far

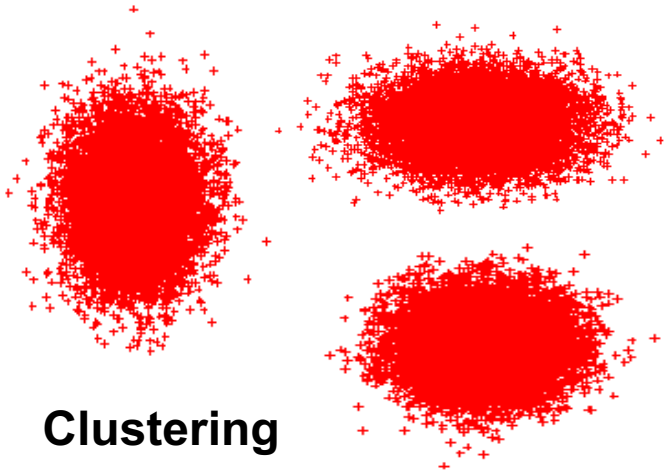
- We need new strategies to deal with big datasets
  - Choosing the right technology is like choosing the right data structure in a program.
- The world of big data is rapidly changing. Being up-to-date is difficult but necessary.

Outline

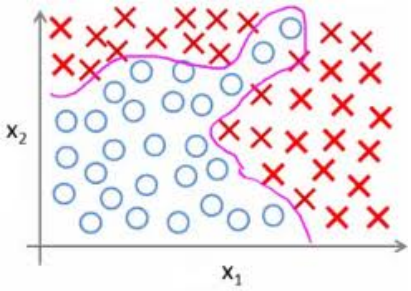
- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Big Data Analytics

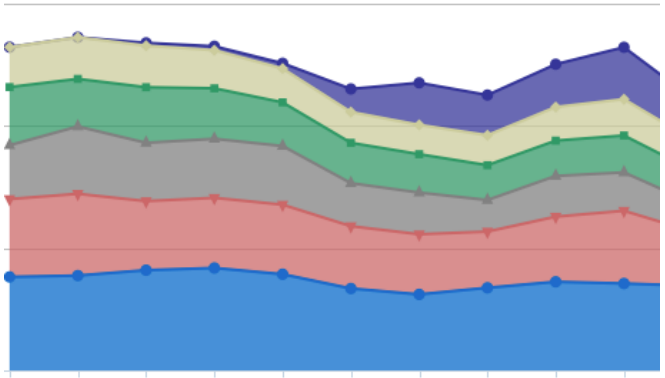
Potential scenarios:



Clustering

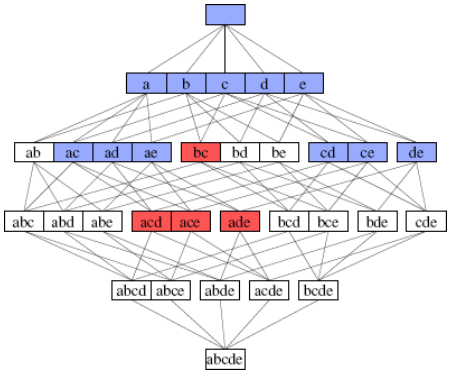


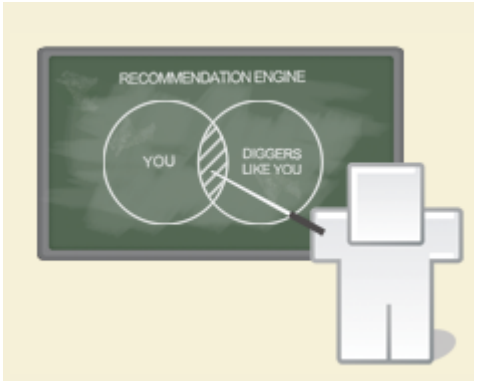
Classification



Real Time Analytics/
Big Data Streams

Association





Recommendation
Systems

Social Media Mining
Social Big Data

Machine learning for Big Data

- Data mining techniques have demonstrated to be very useful tools to extract new valuable knowledge from data.
- The knowledge extraction process from big data has become a very difficult task for most of the **classical** and **advanced** data mining tools.
- The **main challenges** are to deal with:
  - The increasing scale of data
    - at the level of **instances**
    - at the level of **features**
  - The **complexity** of the problem.
  - And many other points

Big Data Analytics: A 3 generational view

Generation	1st Generation	2nd Generation	3rd Generation
Examples	SAS, R, Weka, SPSS, KEEL	Mahout, Pentaho, Cascading	Spark, Haloop, GraphLab, Pregel, Giraph, ML over Storm
Scalability	Vertical	Horizontal (over Hadoop)	Horizontal (Beyond Hadoop)
Algorithms Available	Huge collection of algorithms	Small subset: sequential logistic regression, linear SVMs, Stochastic Gradient Descendent, k-means clustering, Random forest, etc.	Much wider: CGD, ALS, collaborative filtering, kernel SVM, matrix factorization, Gibbs sampling, etc.
Algorithms Not Available	Practically nothing	Vast no.: Kernel SVMs, Multivariate Logistic Regression, Conjugate Gradient Descendent, ALS, etc.	Multivariate logistic regression in general form, k-means clustering, etc. – Work in progress to expand the set of available algorithms
Fault-Tolerance	Single point of failure	Most tools are FT, as they are built on top of Hadoop	FT: HaLoop, Spark Not FT: Pregel, GraphLab, Giraph

Mahout (Samsara)

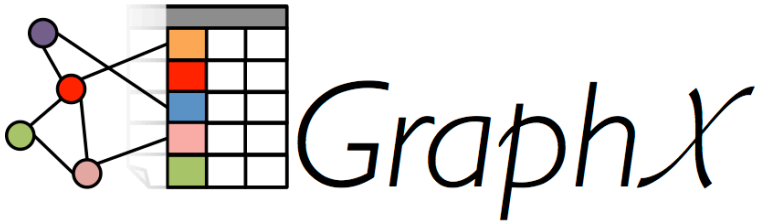
- First ML library initially based on Hadoop MapReduce.
- Abandoned MapReduce implementations from version 0.9.
- Nowadays it is focused on a new **math environment** called **Samsara.**
- It is integrated with **Spark, Flink** and **H2O**
- Main algorithms:
  - Stochastic Singular Value Decomposition (ssvd, dssvd)
  - Stochastic Principal Component Analysis (spca, dspca)
  - Distributed Cholesky QR (thinQR)
  - Distributed regularized Alternating Least Squares (dals)
  - Collaborative Filtering: Item and Row Similarity
  - Naive Bayes Classification

http://mahout.apache.org/

Spark Libraries

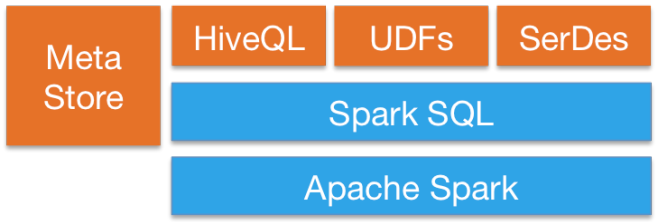
Spark MLlib

https://spark.apache.org/docs/latest/mllib-guide.html



Spark MLlib

Spark SQL



MLlib

Spark MLlib

https://spark.apache.org/docs/latest/mllib-guide.html

Machine Learning Library (MLlib) Guide

MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives, as outlined below:

- Data types
- Basic statistics
  - summary statistics
  - correlations
  - stratified sampling
  - hypothesis testing
  - random data generation
- Classification and regression
  - linear models (SVMs, logistic regression, linear regression)
  - naive Bayes
  - decision trees
  - ensembles of trees (Random Forests and Gradient-Boosted Trees)
  - isotonic regression
- Collaborative filtering
  - alternating least squares (ALS)
- Clustering
  - k-means
  - Gaussian mixture
  - power iteration clustering (PIC)
  - latent Dirichlet allocation (LDA)
  - streaming k-means
- Dimensionality reduction
  - singular value decomposition (SVD)
  - principal component analysis (PCA)
- Feature extraction and transformation
- Frequent pattern mining
  - FP-growth
- Optimization (developer)
  - stochastic gradient descent
  - limited-memory BFGS (L-BFGS)

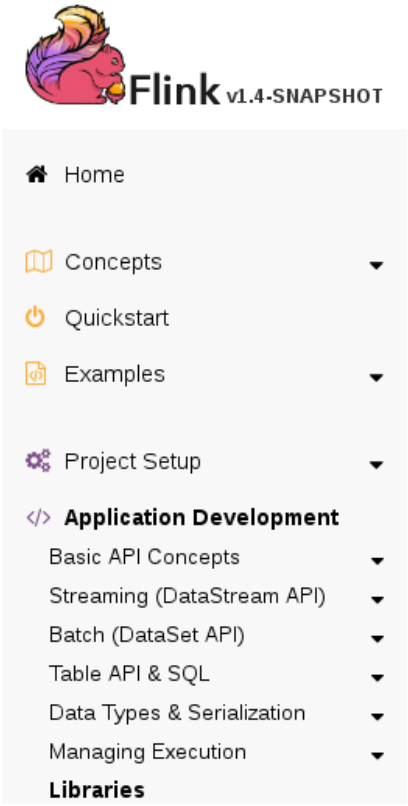
MLlib is under active development. The APIs marked `Experimental/DeveloperApi` may change in future releases, and the migration guide below

http://spark-packages.org/

Spark Packages

A community index of packages for Apache Spark.

FlinkML



https://ci.apache.org/projects/flink/flink-docs-release-1.4/dev/libs/ml/index.html

FlinkML

Supported Algorithms

FlinkML currently supports the following algorithms:

Supervised Learning

- SVM
- Multiple linear regression
- Optimization Framework

Unsupervised Learning

- k-Nearest neighbors join

Data Preprocessing

- Polynomial Features
- Standard Scaler
- MinMax Scaler

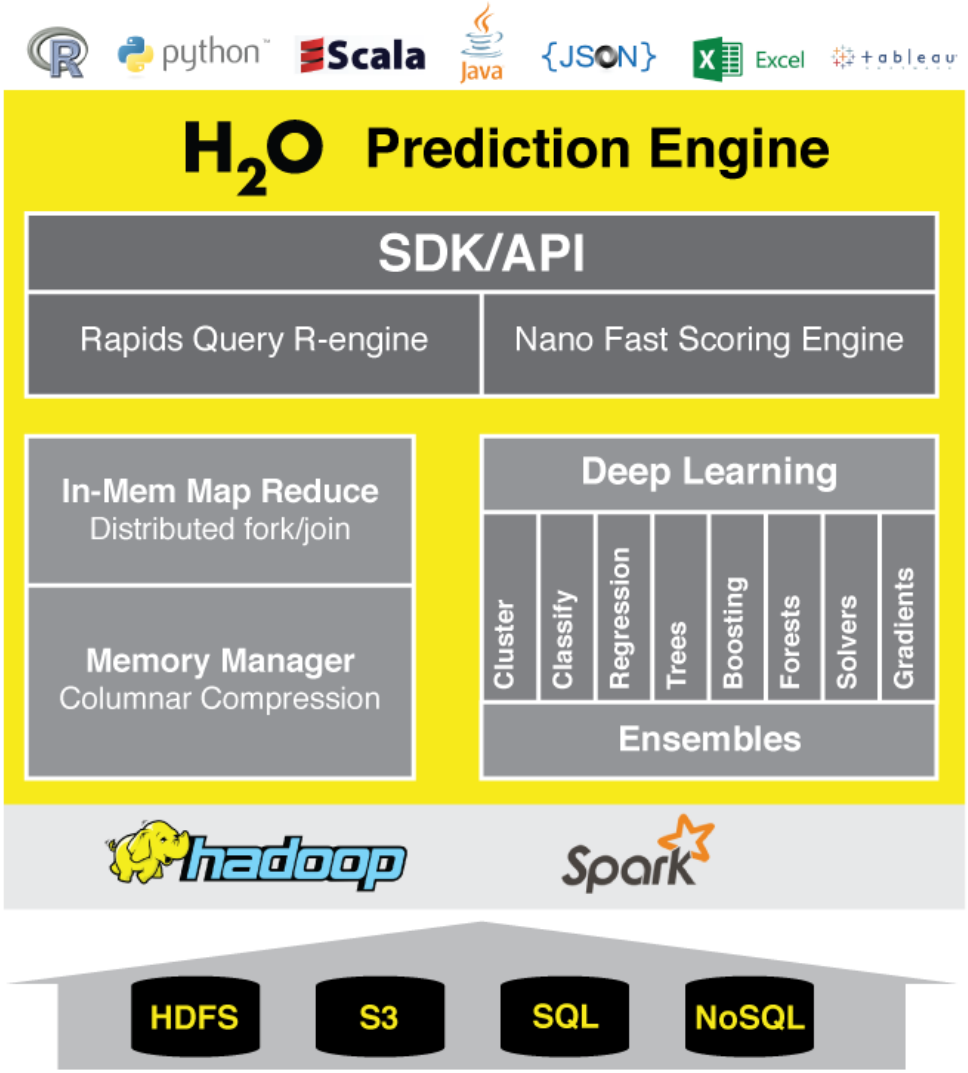
Recommendation

- Alternating Least Squares (ALS)

Utilities

- Distance Metrics
- Cross Validation

H_2O library



http://www.h2o.ai/

Support for R, Python, Hadoop and Spark

- It contains Deep Learning algorithms
  - World record to solve the MNIST problem without preprocessing

http://0xdata.com/blog/2015/02/deep-learning-performance/

Machine Learning in Big Data: Global vs. Local

Two main ways for learning a model in Big Data:

- **Locally**
  - **A model is created for each partition** of the data (only using the data of that partition)
  - **All the models are combined** when predicting the class of a new example → Ensemble
- **Globally**
  - A **single model** is created using **all the available data**
  - They try to obtain the same model as the one that would be obtained if the method could be executed in a single node

Machine Learning in Big Data: Global vs. Local

Local model

- **Advantages**
  - Usually faster
  - Gets faster as the number of partitions is increased
  - Any existing model can be applied
  - Only the aggregation phase has to be designed
- **Disadvantages**
  - **Slow in test phase**, too many models have to be executed
  - **Loss of accuracy** as the number of partitions increases
    - With few partitions, accuracy can improve due to the ensemble effect
    - With too many partitions, the accuracy tends to drop, since there are not enough examples in each partition
  - **They do not take advantage of the data as a whole**

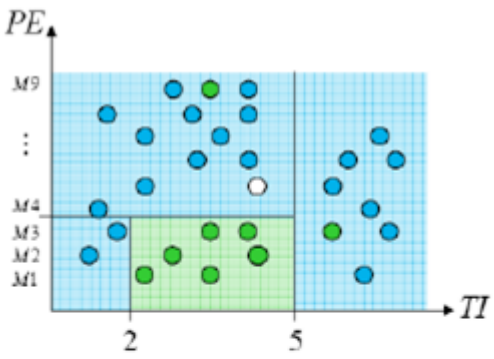
Machine Learning in Big Data: Global vs. Local

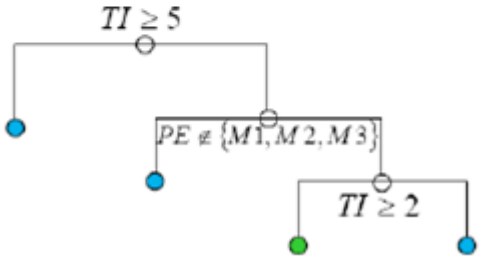
Global model

- **Advantages**
  - Greater **accuracy** is expected (not proved)
  - All the examples are used to learn a single model
  - Anyway, a global ensemble can also be built
  - The model is independent of the number of partitions
  - Faster in test phase
- **Disadvantages**
  - **More complex design and implementation**
  - Distributed nature of Big Data processing has to be taken into account (computation/communication)

Case of Study: Random Forests

- Random Forest is a very well-known machine learning technique for **classification** or regression.
  - Ensemble learning
  - Tree-based models
  - Random selection of features

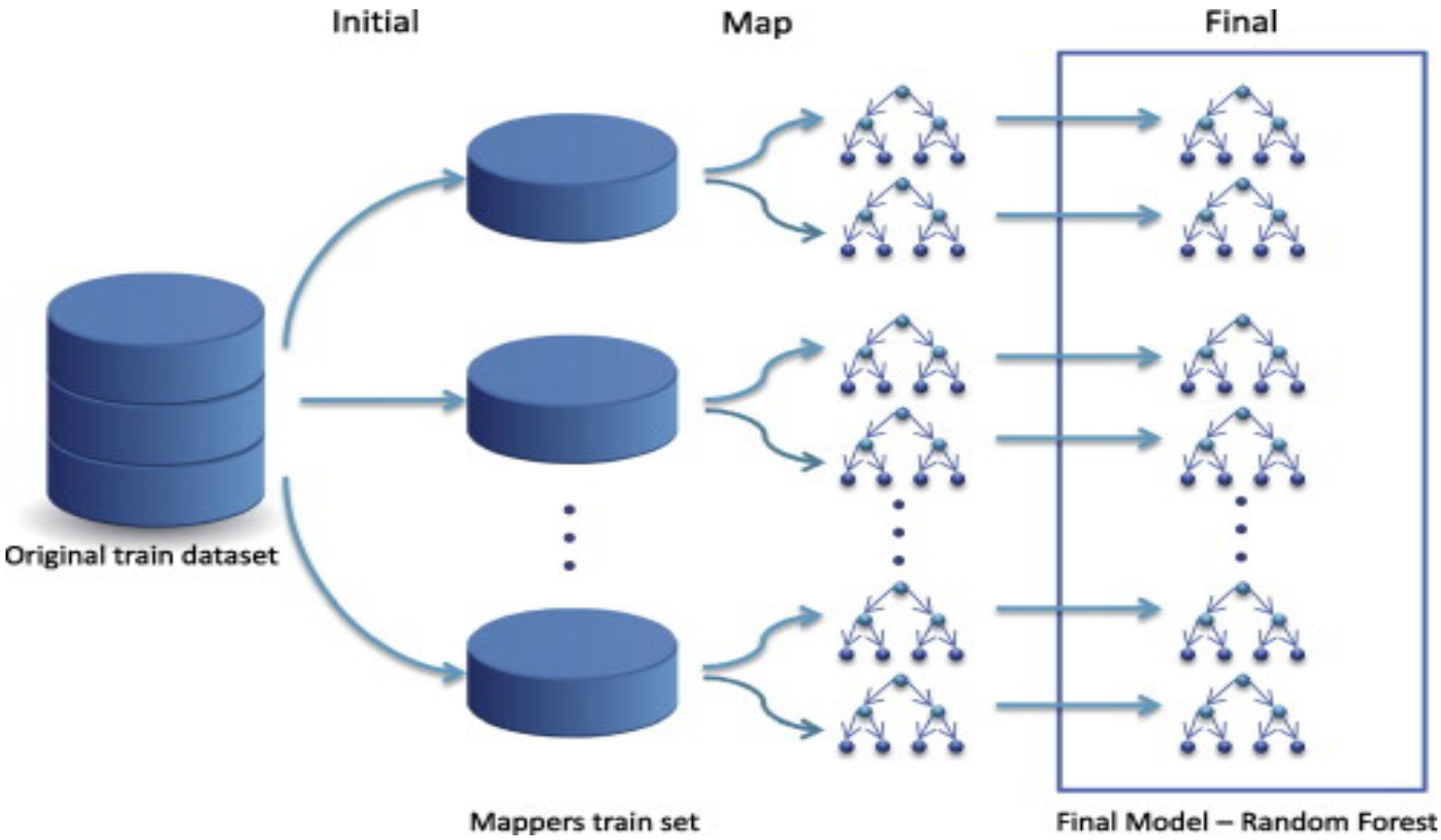




- Most promising characteristics:
  - Great generalization capabilities
  - Detect variable importance
  - Relatively efficient on large data bases.

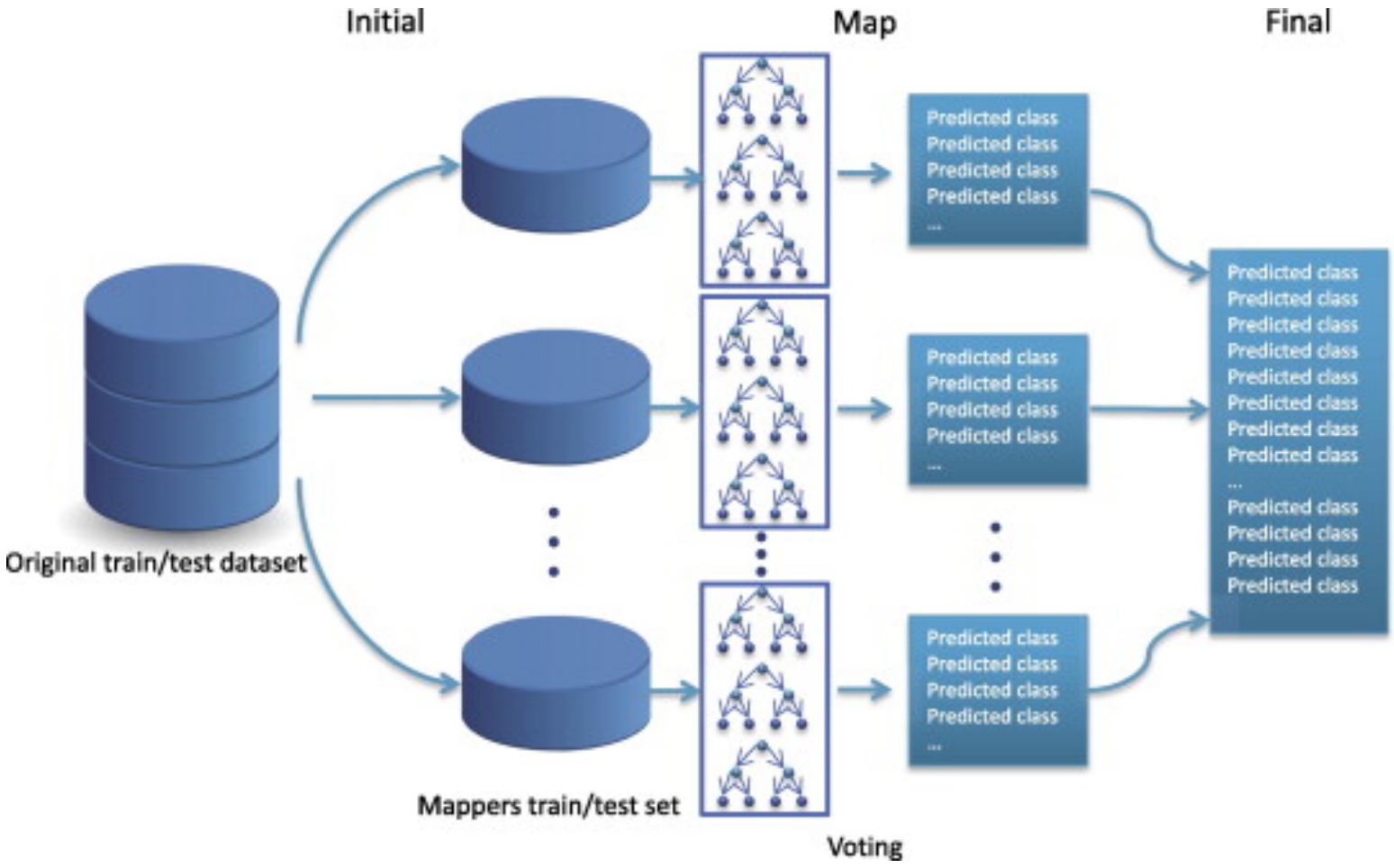
Random Forest under MapReduce

1) Building phase

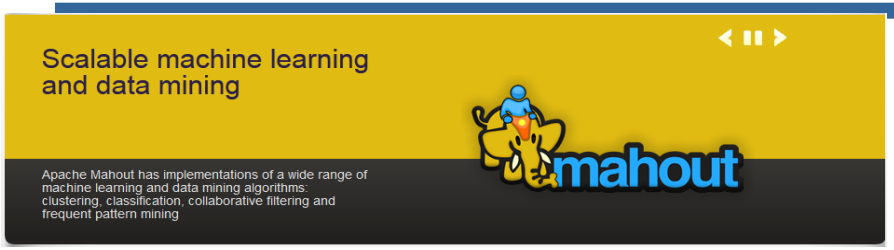


Random Forest under MapReduce

2) Testing phase



Random Forest under MapReduce



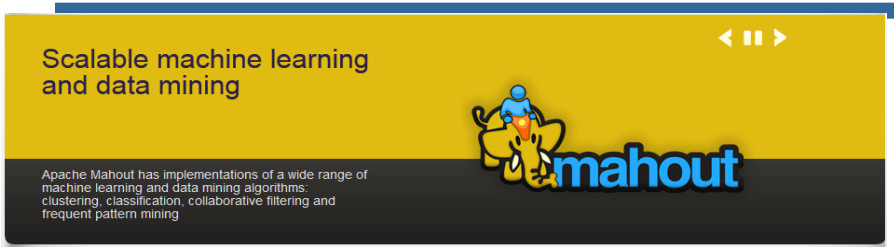
Case of Study: Random Forest for KddCup'99

Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

Time elapsed (seconds) for sequential versions:

Datasets	RF		
	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC
DOS_versus_R2L	4454.58	28073.79	NC
DOS_versus_U2R	3848.97	24774.03	NC
normal_versus_PRB	468.75	6011.70	NC
normal_versus_R2L	364.66	4773.09	14703.55
normal_versus_U2R	295.64	4785.66	14635.36

Random Forest under MapReduce



Case of Study: Random Forest for KddCup'99

Class	Instance Number
normal	972.781
DOS	3.883.370
PRB	41.102
R2L	1.126
U2R	52

	10%	50%	full
DOS_versus_normal	6344.42	49134.78	NC
DOS_versus_PRB	4825.48	28819.03	NC

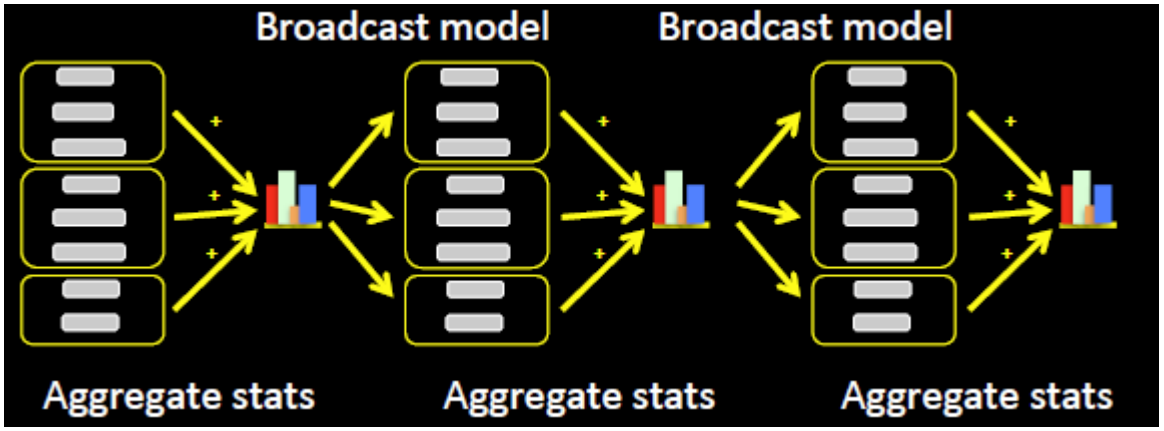
Time elapsed (seconds) for Big data versions with 20 partitions:

Datasets	RF-BigData		
	10%	50%	full
DOS_versus_normal	98	221	236
DOS_versus_PRB	100	186	190
DOS_versus_R2L	97	157	136
DOS_versus_U2R	93	134	122
normal_versus_PRB	94	58	72
normal_versus_R2L	92	39	69
normal_versus_U2R	93	52	64

Cluster ATLAS: 16 nodes
-Microprocessors: 2 x Intel E5-2620 (6 cores/12 threads, 2 GHz)
- RAM 64 GB DDR3 ECC 1600MHz
- Mahout version 0.8

Case of Study: (Global) Decision Trees for Big Data

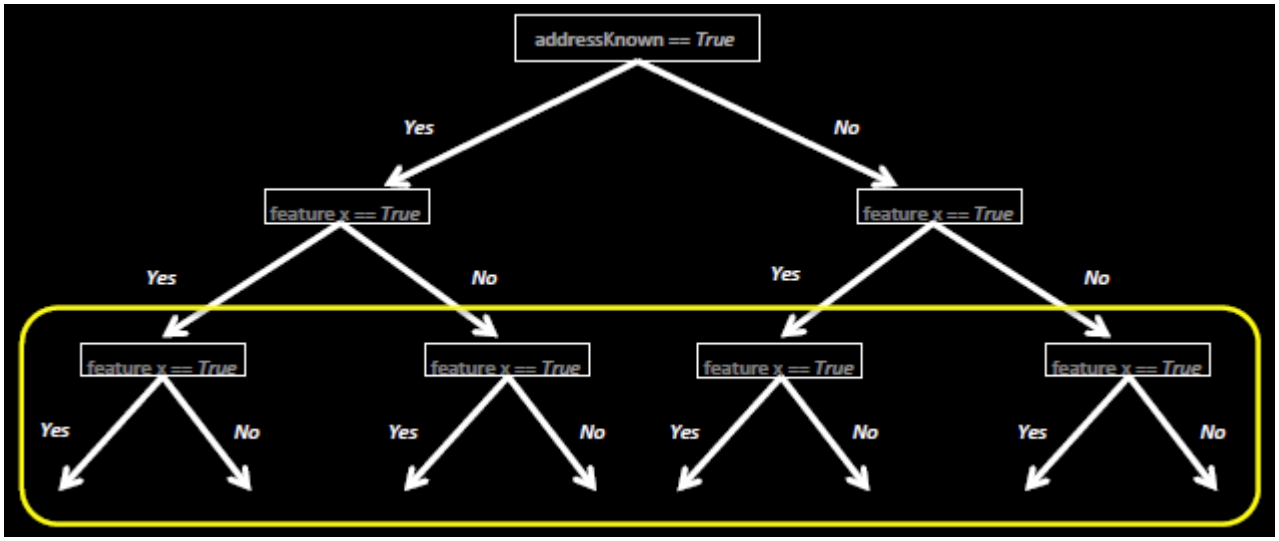
- **Decision Trees in Spark**
  - **Differences** with respect to classical models
    - All the nodes in a level are learned with a single pass through the whole dataset
    - Numeric attributes are discretized into bins in order to reduce the computational cost



https://speakerdeck.com/jkbradley/mllib-decision-trees-at-sf-scala-baml-meetup

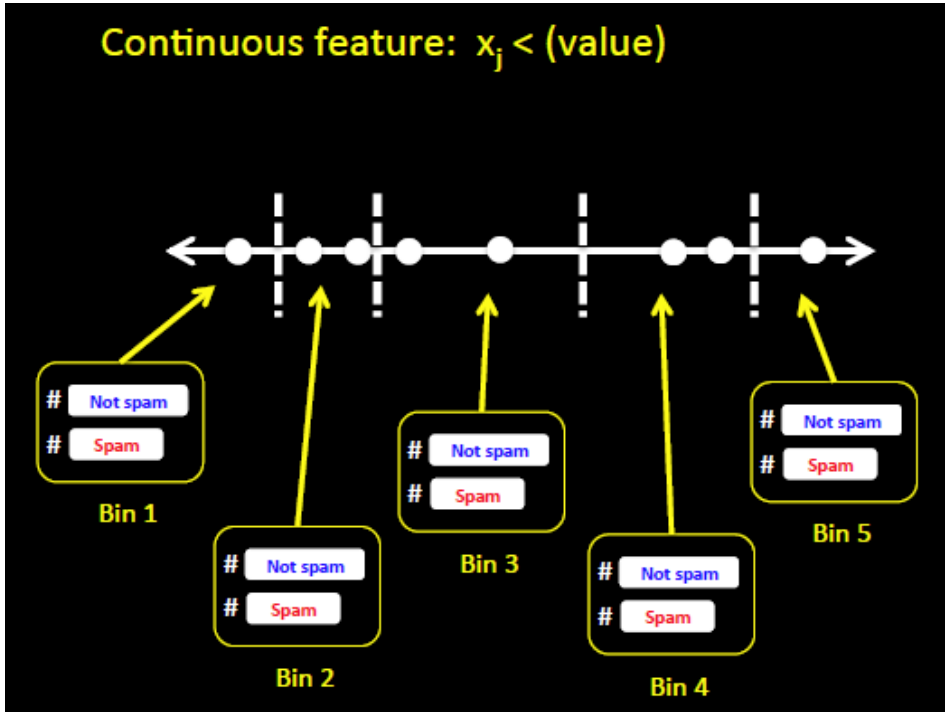
Global Decision Trees for Big Data

- **Decision Trees in Spark**
  - **Differences** with respect to classical models
    - All the nodes in a level are learned with a single pass through the whole dataset

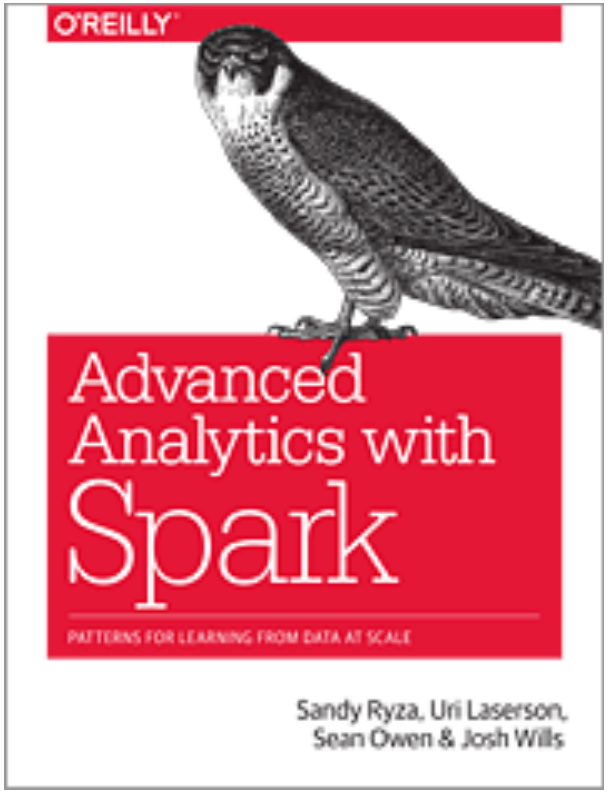


Global Decision Trees for Big Data

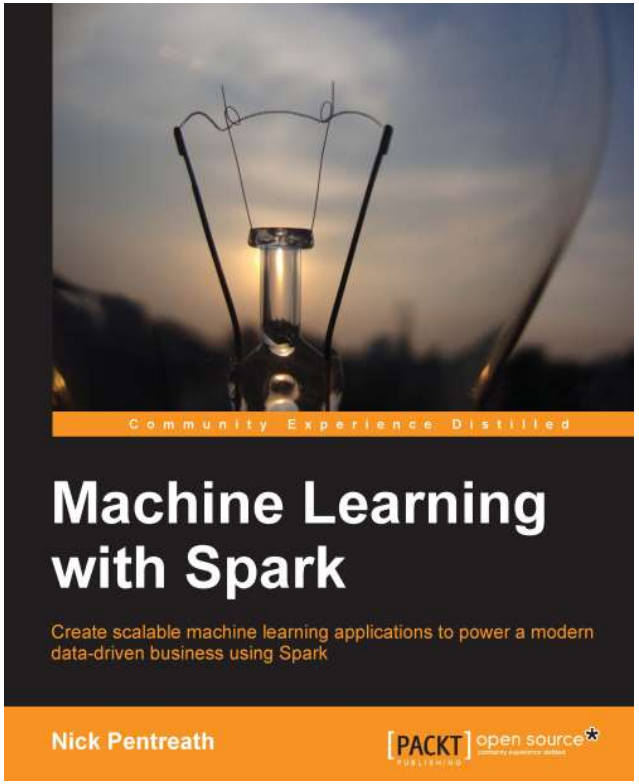
- **Decision Trees in Spark**
  - **Differences** with respect to classical models
    - Numeric attributes are discretized into bins in order to reduce the computational cost



Big Data Analytics: 2 books



9 cases of study



10 chapters giving a quick glance on Machine Learning with Spark

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Fuzzy-based models for Big Data learning

BIG DATA with Fuzzy Models

OUTLINE

- Data fragmentation and lack of data problem
- Big Data classification with fuzzy models
- Are FRBCSs robust with respect to the lack of data?
- Conclusions and future challenges

FUZZ-IEEE 2017 NAPLES, ITALY

BIG DATA

Fuzzy-based models for Big Data learning

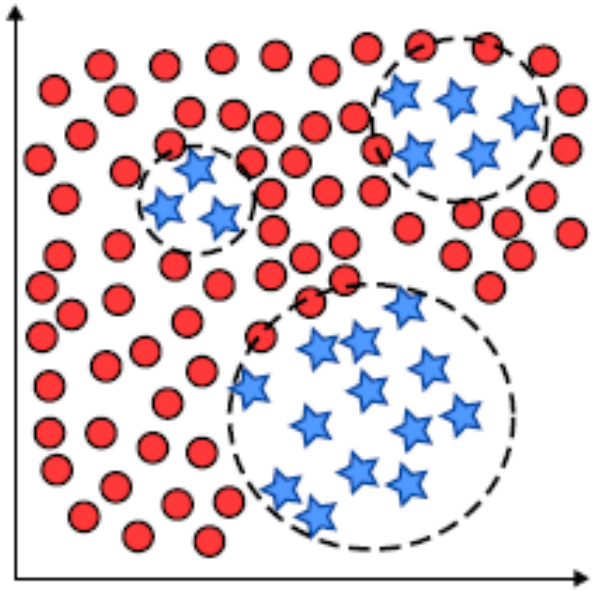
BIG DATA with Fuzzy Models

- OUTLINE
  - Data fragmentation and lack of data problem
  - Big Data classification with fuzzy models
  - Are FRBCSs robust with respect to the lack of data?
  - Conclusions and future challenges

FUZZ-IEEE 2017 NAPLES, ITALY

BIG DATA

Big Data Classification: Data Fragmentation and Lack of Data Problem



(b) Small disjuncts

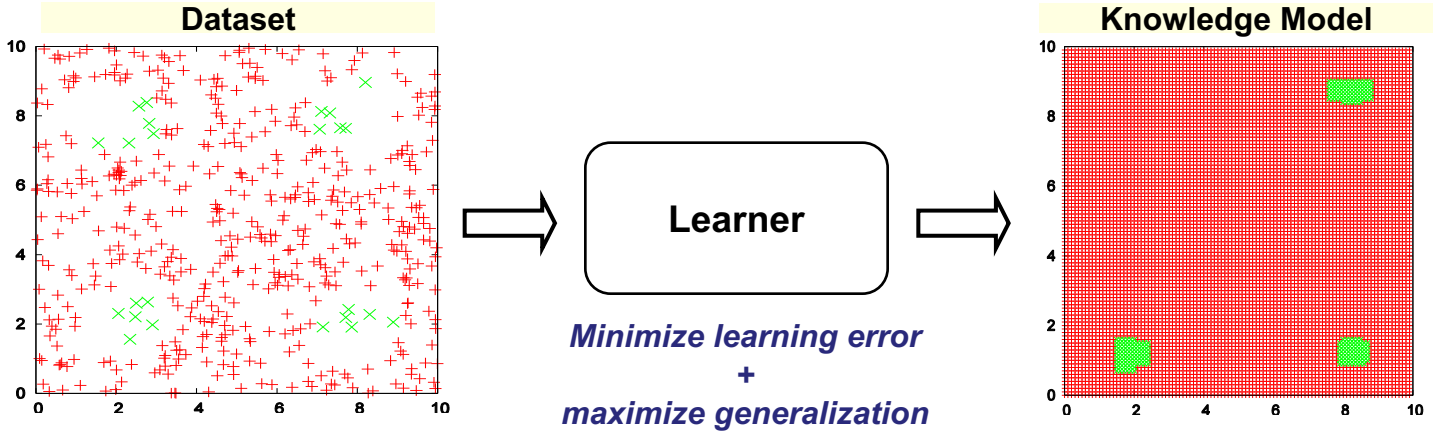
Rare cases or Small disjuncts are those disjuncts in the learned classifier that cover few training examples.

T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

G.M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations 6:1 (2004) 7-19

Big Data Classification: Data Fragmentation and Lack of Data Problem

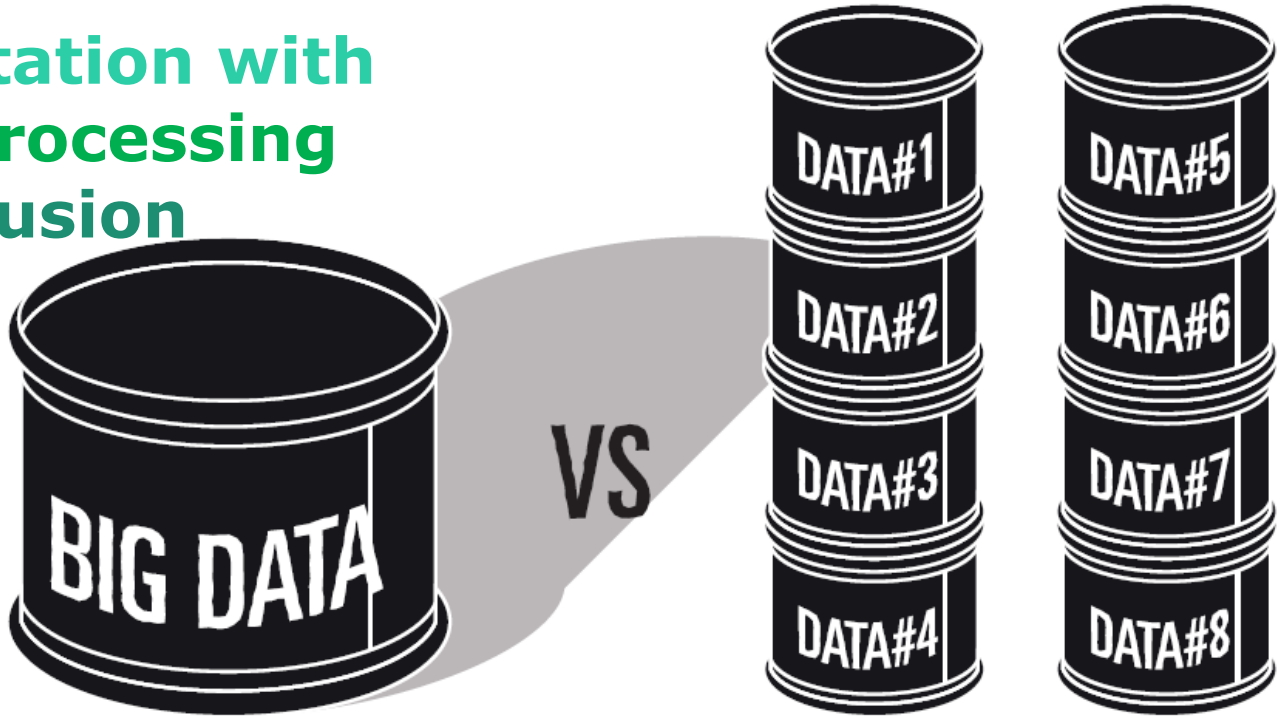
It becomes very hard for the learning algorithm to obtain a model that is able to perform a good generalization when there is not enough data that represents the boundaries of the problem.



Big Data Classification: Data Fragmentation and Lack of Data Problem

MapReduce

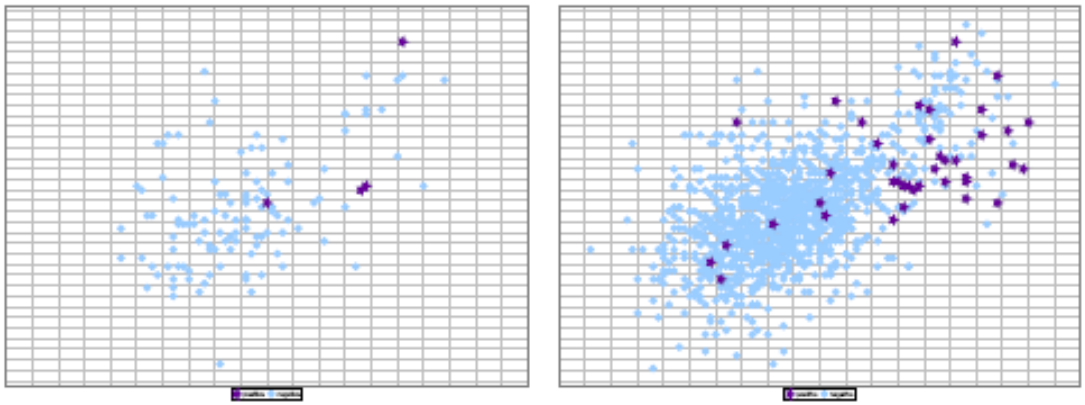
Data Fragmentation with Parallel Processing + Model fusion



Small disjuncts arise with MapReduce data fragmentation. This problem is **accentuated for imbalance** classification: **Lack of Data/lack of density between classes**

Big Data Classification: Data Fragmentation and Lack of Data Problem

Data fragmentation - Lack of data



(a) 10 % of training instances (b) 100 % of training instances

Figure 11: Lack of density or small sample size on the yeast4 dataset

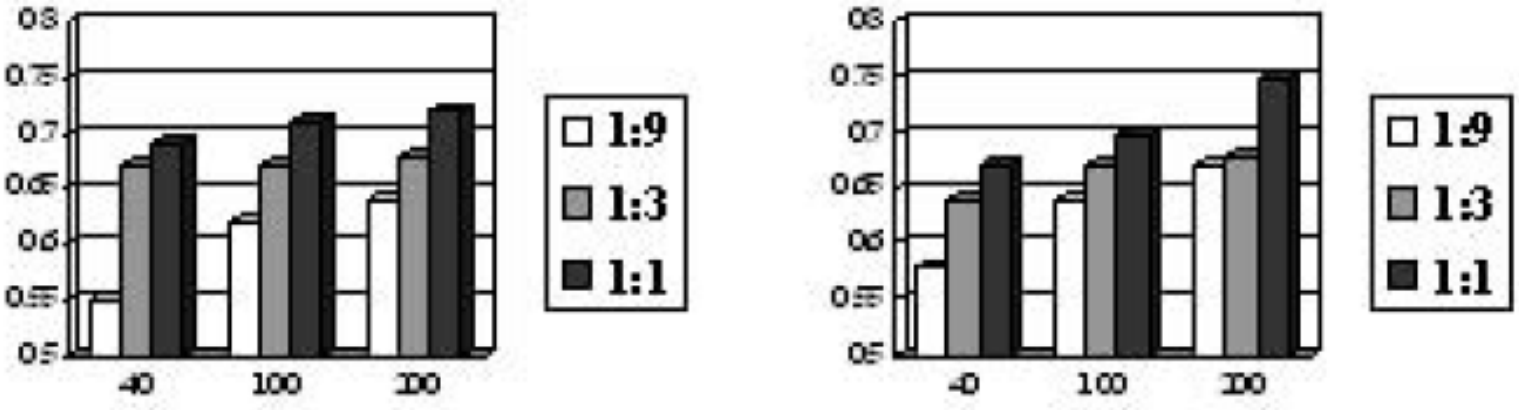
The lack of data in the training data may cause the introduction of small disjuncts.

What it is also most significant, when the concentration of minority examples is so low that they can be simply treated as noise.

Big Data Classification: Data Fragmentation and Lack of Data Problem

Lack of data

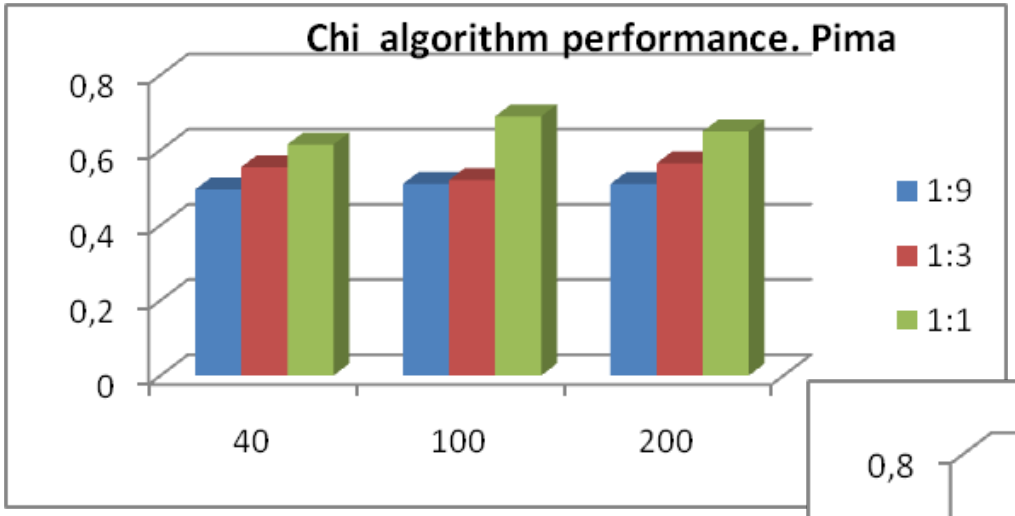
Left-C4.5, right-Backpropagation (Pima data set): These results show that the performance of classifiers is repaired as the training set size increases.
This suggests that small disjuncts play a role in the performance loss of class imbalanced domains.

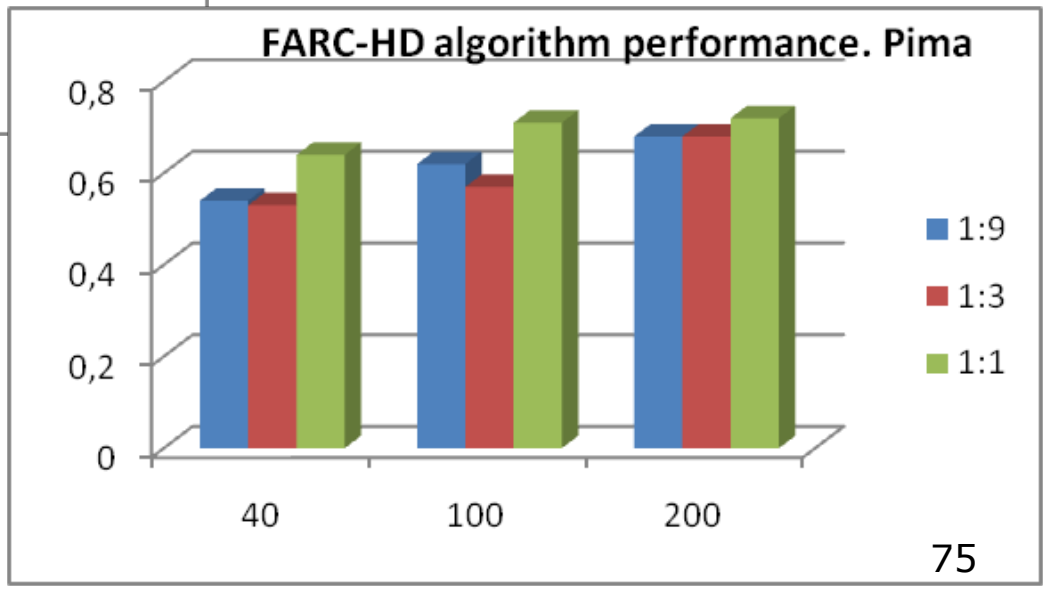


T. Jo, N. Japkowicz. Class imbalances versus small disjuncts. SIGKDD Explorations 6:1 (2004) 40-49

Big Data Classification: Data Fragmentation and Lack of Data Problem

Lack of data. Fuzzy models performance





Robustness to the lack of data?

Fuzzy-based models for Big Data learning

- OUTLINE
  - Data fragmentation and lack of data problem
  - Big Data classification with fuzzy models
  - Are FRBCSs robust with respect to the lack of data?
  - Conclusions and future challenges

Uncertainty and Big Data

- Uncertainty is inherent to Big Data due to
  - Heterogeneous sources
  - Variety in data
  - Incomplete data
  - Veracity in question
- Fuzzy Rule Based Classification Systems can manage
  - Uncertainty
  - Vagueness
  - Lack of data/Data fragmentation

Linguistic Fuzzy Models as a robust solution for Big Data

- Linguistic FRBCSs have been used in distributed environments for a long time
- In the context of Big Data applications MapReduce is the "key":
  - simple,
  - fault-tolerant,
  - scalable.
- The programming framework differs from traditional schemes:
  - Map tasks imply the building of local / partial models
  - Reduce tasks aggregate partial models

Big Data Classification with Fuzzy Models

Chi-FRBCS-BigData: A Case of Study

We choose a simple Learning Method to analyze the potential of FRBCSs for Big Data Classification

- MapReduce design based on the Chi et al. FRBCS algorithm (two different MapReduce processes)
  - Phase 1: Building the Fuzzy Rule Base
  - Phase 2: Estimating the class of samples belonging to Big Data sample sets
- Two versions that differ in the *Reduce function* of the building of the Fuzzy RB are considered
  - Chi-FRBCS-BigData-Max
  - Chi-FRBCS-BigData-Average

S. Río, V. López, J.M. Benítez, F. Herrera. *A MapReduce Approach to Address Big Data Classification Problems Based on the Fusion of Linguistic Fuzzy Rules.* International Journal of Computational Intelligence Systems 8:3 (2015) 422-437. doi: 10.1080/18756891.2015.1017377

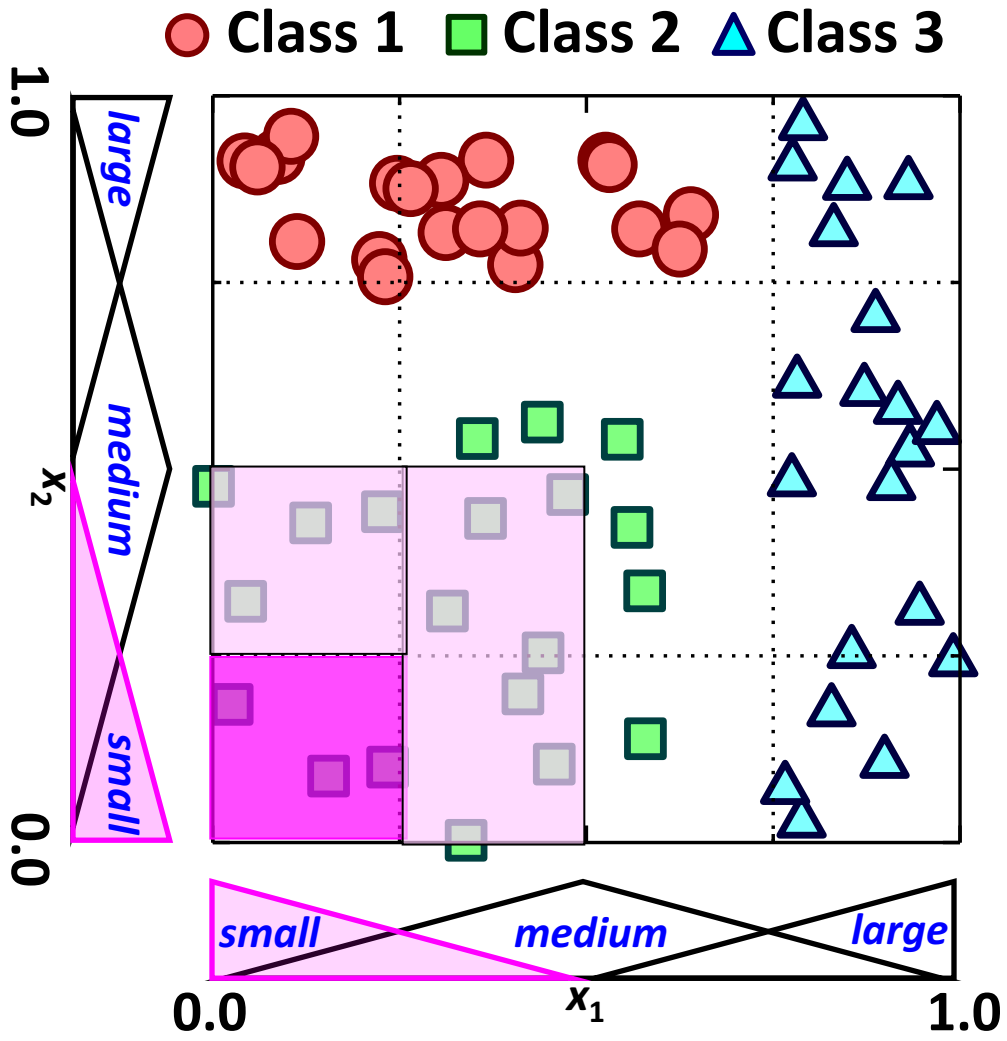
Big Data Classification with Fuzzy Models

Chi-FRBCS

- Generates rules as "**Rule R_j:** IF x_1 IS A^1_j AND … AND x_n IS A^n_j THEN Class = C_j with RW_j"
- Builds the fuzzy partition using equally distributed triangular membership functions
- Builds the RB creating a fuzzy rule associated to each example
- Rules with the same antecedent may be created:
  - Same consequent → Delete duplicated rules
  - Different consequent → Preserve highest weight rule

Z. Chi, H. Yan and T. Pham, Fuzzy algorithms with applications to image processing and pattern recognition, World Scientific, 1996.

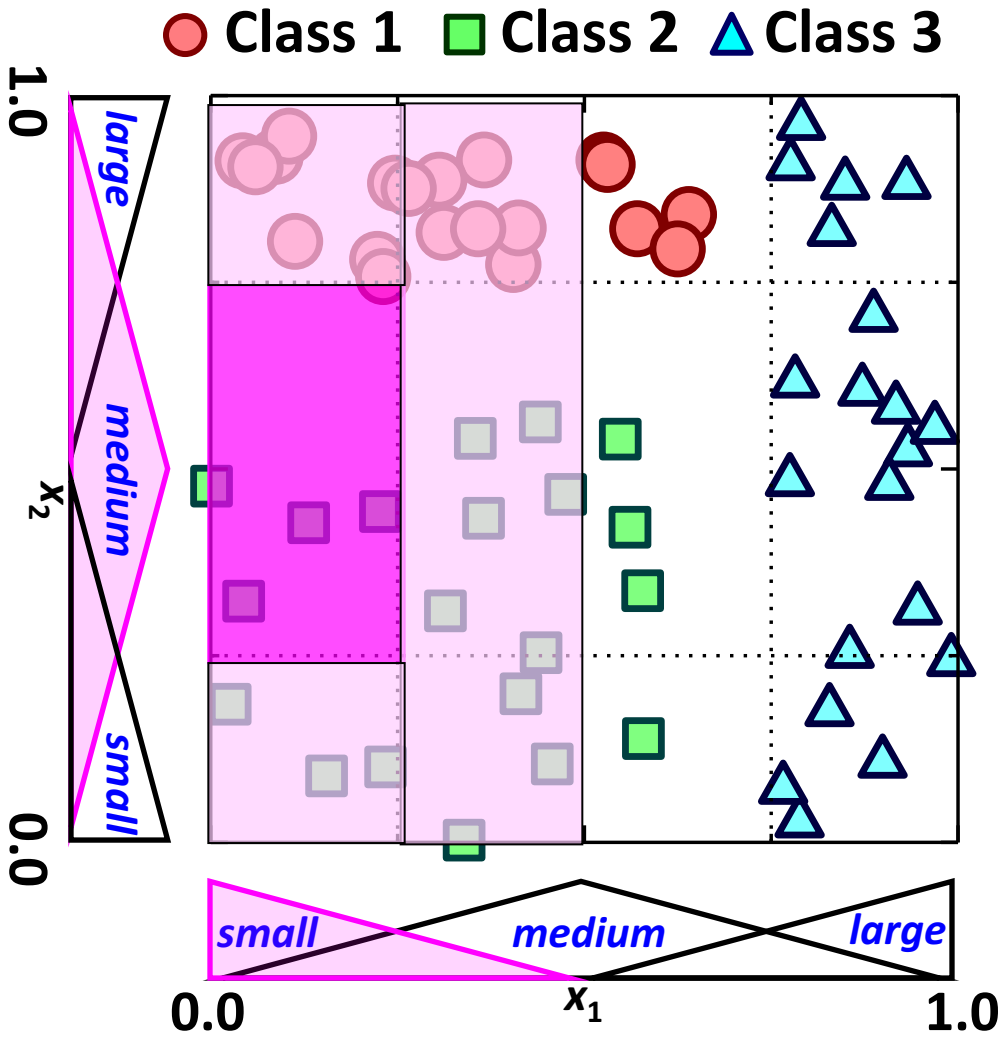
Big Data Classification with Fuzzy Models



Rule Base (No Weights):

If x_1 is *small* and x_2 is *small* then Class 2

Big Data Classification with Fuzzy Models

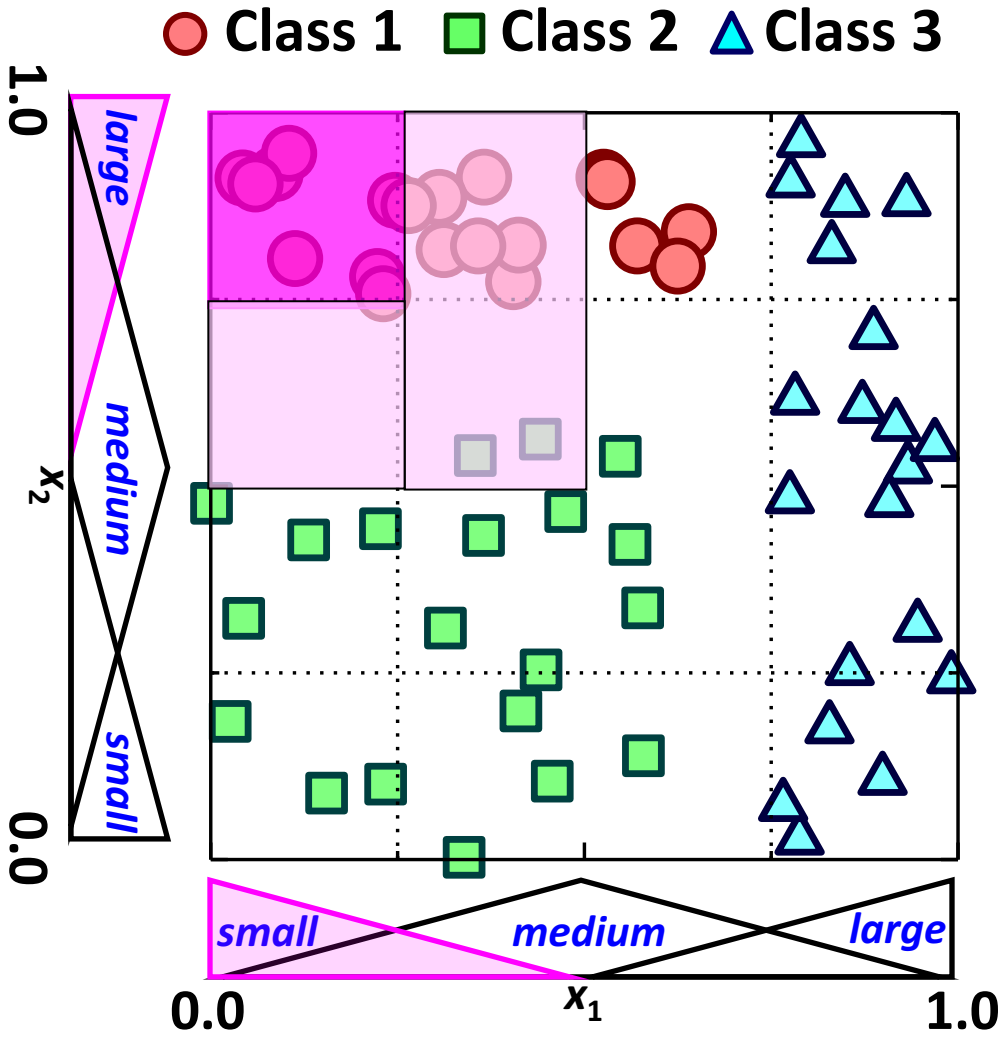


Rule Base (No Weights)

If x_1 is *small* and x_2 is *small* then Class 2

If x_1 is *small* and x_2 is *medium* then Class 2

Big Data Classification with Fuzzy Models



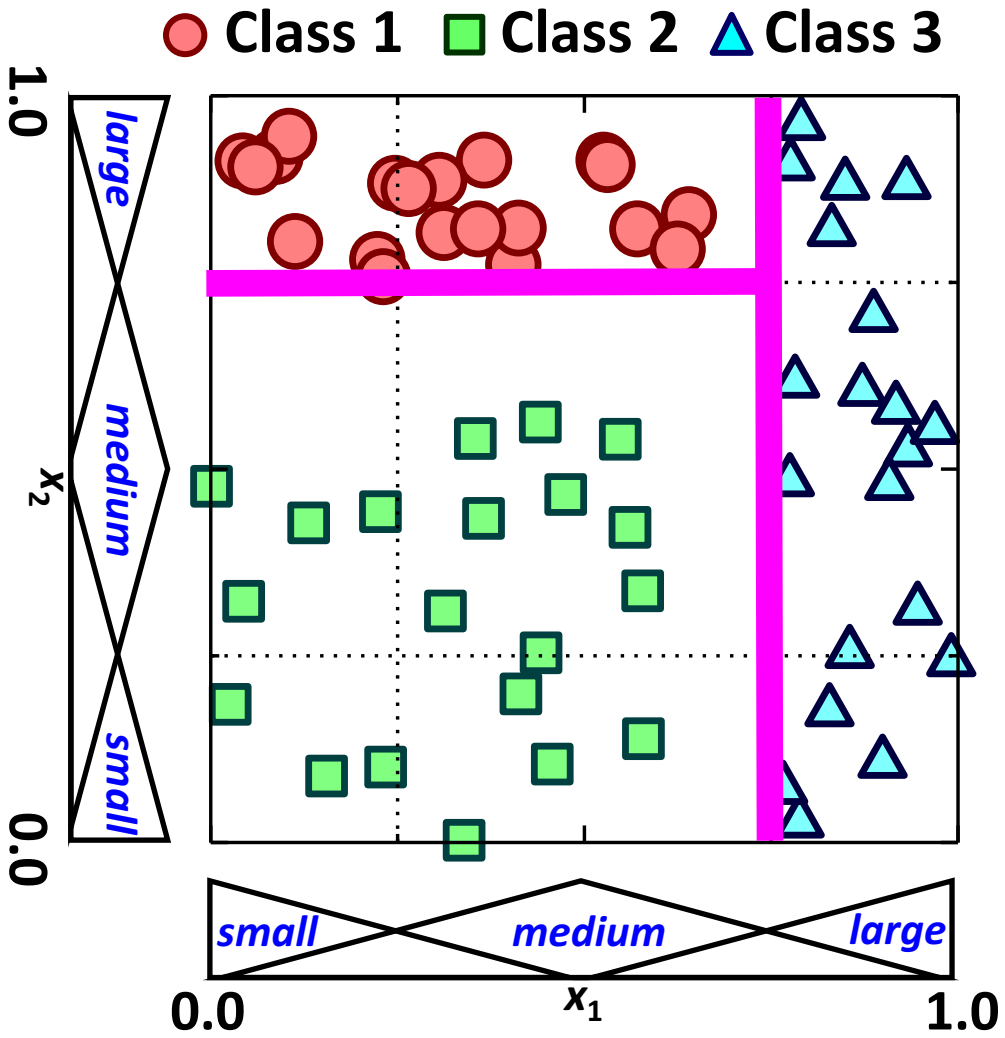
Rule Base (No weights)

If x_1 is *small* and x_2 is *small*
then Class 2

If x_1 is *small* and x_2 is *medium*
then Class 2

If x_1 is *small* and x_2 is *large*
then Class 1

Big Data Classification with Fuzzy Models



Classification Boundaries

If x_1 is *small* and x_2 is *small*
then Class 2

If x_1 is *small* and x_2 is *medium*
then Class 2

If x_1 is *small* and x_2 is *large*
then Class 1

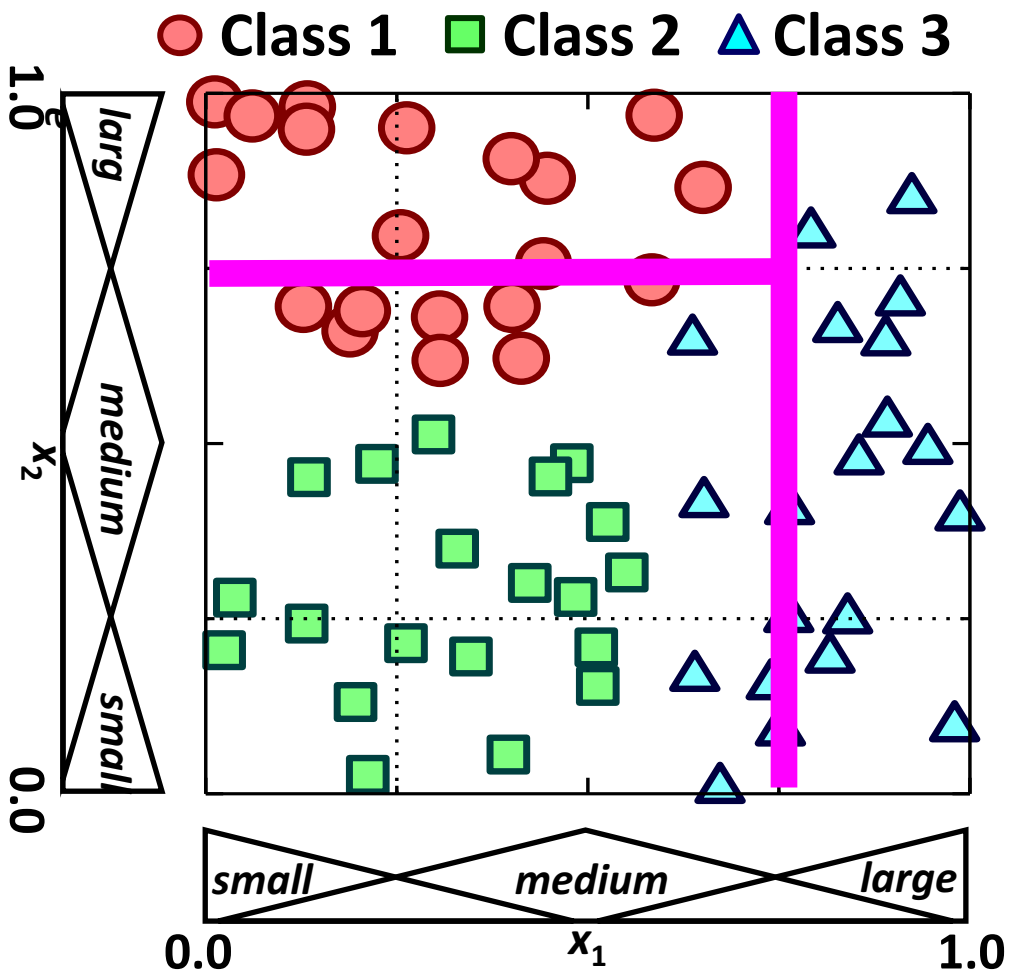
. . .

If x_1 is *large* and x_2 is *large*
then Class 3

High interpretability

Big Data Classification with Fuzzy Models

Standard rules (w/o weights) does not always get a good representation



Rule Base (No Weights)

If x_1 is *small* and x_2 is *small*
then Class 2

If x_1 is *small* and x_2 is *medium*
then Class 2

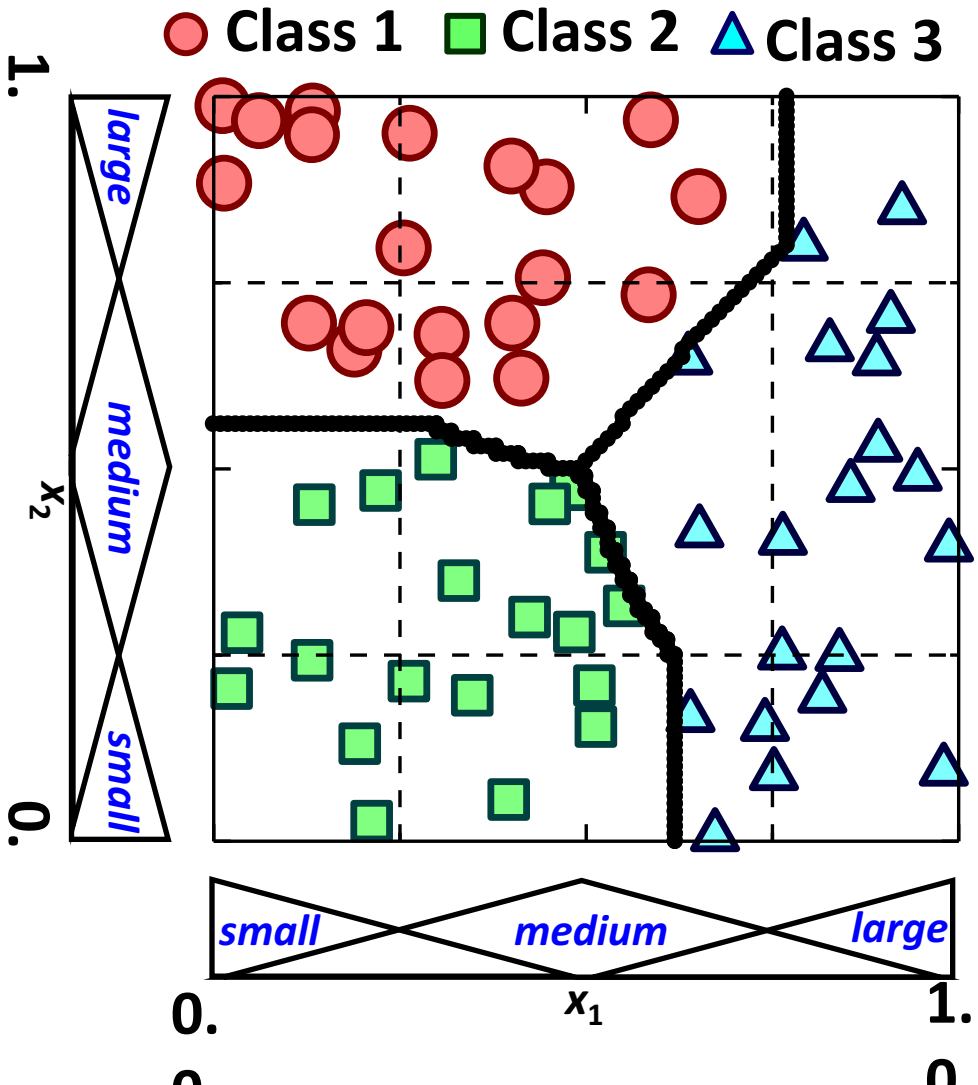
If x_1 is *small* and x_2 is *large*
then Class 1

. . .

If x_1 is *large* and x_2 is *large*
then Class 3

High Interpretability
Low Accuracy

Big Data Classification with Fuzzy Models



RB (No weights)

If x_1 is *small* and x_2 is *medium*

then Class 2

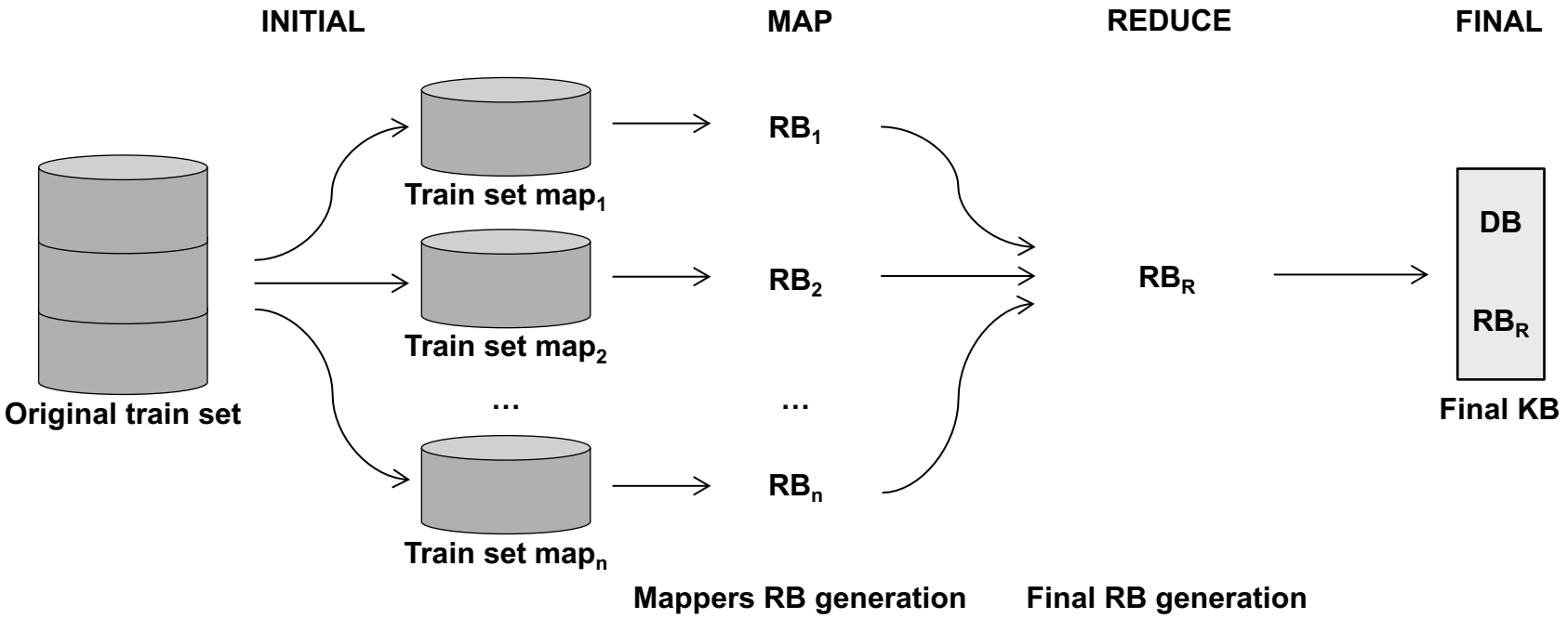
RB (Weights == Certainty Factor)

If x_1 is *small* and x_2 is *medium*

then Class 2 with 0.158

Big Data Classification with Fuzzy Models

Building the RB with Chi-FRBCS-BigData: A Map Reduce approach



The key of a MapReduce data partitioning approach is usually on the reduce phase

Two alternative reducers (Max vs average weights)

Building the FRB with Chi-FRBCS-BigData-Max

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.8743$
R_2: IF $A_1 = L_2$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.9142$
…

RB_1

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.9254$
R_2: IF $A_1 = L_1$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.8842$
…

RB_2

R_1: IF $A_1 = L_2$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.6534$
R_2: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.7142$
…

RB_3

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_2; $RW_1 = 0.2143$
R_2: IF $A_1 = L_3$ AND $A_2 = L_2$ THEN C_2; $RW_3 = 0.4715$
…

…

RB_4

R_1: IF $A_1 = L_2$ AND $A_2 = L_3$ THEN C_2; $RW_3 = 0.7784$
R_2: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_2 = 0.8215$
…

RB_n

REDUCE

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.9254$
R_2: IF $A_1 = L_2$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.9142$
R_3: IF $A_1 = L_1$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.8842$
R_4: IF $A_1 = L_2$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.6534$
R_5: IF $A_1 = L_3$ AND $A_2 = L_2$ THEN C_2; $RW_3 = 0.4715$
R_6: IF $A_1 = L_2$ AND $A_2 = L_3$ THEN C_2; $RW_3 = 0.7784$
…

RB_R

Final RB generation

RB_1, R_1, C_1, RW = 0.8743
RB_2, R_1, C_2, RW = 0.9254
RB_3, R_2, C_1, RW = 0.7142
RB_4, R_1, C_2, RW = 0.2143
RB_5, R_2, C_1, RW = 0.8215

Building the FRB with Chi-FRBCS-BigData-Ave

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.8743$
R_2: IF $A_1 = L_2$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.9142$
…

RB_1

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.9254$
R_2: IF $A_1 = L_1$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.8842$
…

RB_2

R_1: IF $A_1 = L_2$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.6534$
R_2: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.7142$
…

RB_3

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_2; $RW_1 = 0.2143$
R_2: IF $A_1 = L_3$ AND $A_2 = L_2$ THEN C_2; $RW_3 = 0.4715$
…

RB_4

…

R_1: IF $A_1 = L_2$ AND $A_2 = L_3$ THEN C_2; $RW_3 = 0.7784$
R_2: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_2 = 0.8215$
…

RB_n

REDUCE

R_1: IF $A_1 = L_1$ AND $A_2 = L_1$ THEN C_1; $RW_1 = 0.8033$
R_2: IF $A_1 = L_2$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.9142$
R_3: IF $A_1 = L_1$ AND $A_2 = L_2$ THEN C_2; $RW_2 = 0.8842$
R_4: IF $A_1 = L_2$ AND $A_2 = L_1$ THEN C_2; $RW_3 = 0.6534$
R_5: IF $A_1 = L_3$ AND $A_2 = L_2$ THEN C_2; $RW_3 = 0.4715$
R_6: IF $A_1 = L_2$ AND $A_2 = L_3$ THEN C_2; $RW_3 = 0.7784$
…

RB_R

Final RB generation

RB_1, R_1, C_1, RW = 0.8743
RB_2, R_1, C_2, RW = 0.9254
RB_3, R_2, C_1, RW = 0.7142
RB_4, R_1, C_2, RW = 0.2143
RB_5, R_2, C_1, RW = 0.8215

→

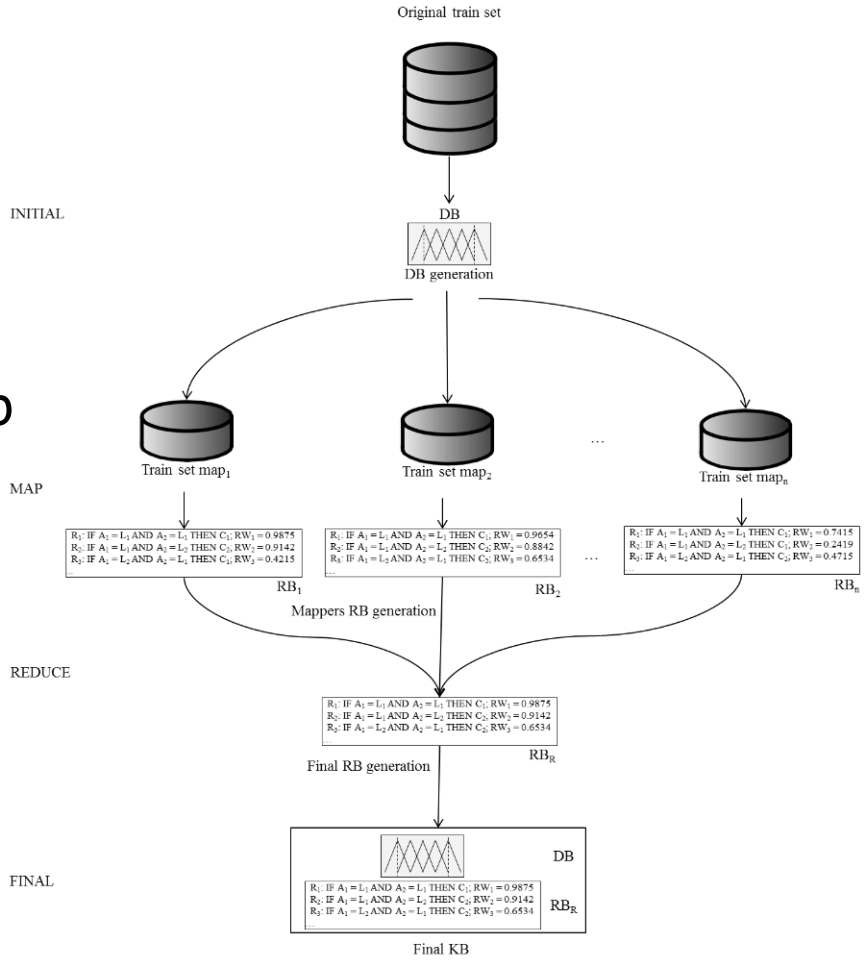
RC_1, C_1, $RW_{ave} = 0.8033$
RC_2, C_2, $RW_{ave} = 0.5699$

Summary of Chi-FRBCS-BigData

Homegenous fuzzy partitions shared by all Map tasks

Rule with equal antecedents are merged.

RWs are averaged



The original Chi fuzzy rule learning algorithm is applied

Consequent by P-CF

S. Río, et al. A MapReduce Approach to Address Big Data Classification Problems Based on the Fusion of Linguistic Fuzzy Rules. International Journal of Computational Intelligence Systems 8:3 (2015) 422-437.

Big Data Classification with Fuzzy Models

Experimental Analysis: Chi-FRBCS-BigData

- 6 Datasets with two classes problem
- Stratified 10 fold cross-validation
- Parameters:
  - Conjunction Operator: Product T-norm
  - Rule Weight: Penalized Certainty Factor
  - Number of fuzzy labels per variable: 3 labels

Experimental Framework

Datasets	#Ex.	#Atts.	Selected classes	#Samples per class
RLCP	5749132	2	(FALSE; TRUE)	(5728201; 20931)
Kddcup_DOS_vs_normal	4856151	41	(DOS; normal)	(3883370; 972781)
Poker_0_vs_1	946799	10	(0; 1)	(513702; 433097)
Covtype_2_vs_1	495141	54	(2; 1)	(283301; 211840)
Census	141544	41	(-_50000.; 50000+.)	(133430; 8114)
Fars_Fatal_Inj_vs_No_Inj	62123	29	(Fatal_Inj; No_Inj)	(42116; 20007)

Big Data Classification with Fuzzy Models

Analysis of the Performance, Precision

Datasets	8 maps					
	Chi-FRBCS		Chi-BigData-Max		Chi-BigData-Ave	
	Acc_{tr}	Acc_{tst}	Acc_{tr}	Acc_{tst}	Acc_{tr}	Acc_{tst}
Poker_0_vs_1	63.72	**61.77**	62.93	60.74	63.12	60.91
Covtype_2_vs_1	74.65	74.57	74.69	**74.63**	74.66	74.61
Census	96.52	86.06	97.12	**93.89**	97.12	93.86
Fars_Fatal_Inj_vs_No_Inj	99.66	89.26	97.01	95.07	97.18	**95.25**
Average	83.64	77.92	82.94	81.08	83.02	**81.16**

Good precision!

Big Data Classification with Fuzzy Models

BIG DATA with Fuzzy Models

Analysis of the Performance, Precision

Datasets	16 mappers			
	Chi-BigData-Max		Chi-BigData-Ave	
	Acc_{tr}	Acc_{tst}	Acc_{tr}	Acc_{tst}
RLCP	99.63	99.63	99.63	99.63
Kddcup_DOS_vs_normal	99.93	99.93	99.93	99.93
Poker_0_vs_1	62.18	59.88	62.58	60.35
Covtype_2_vs_1	74.77	74.72	74.77	74.69
Census	97.14	93.75	97.15	93.52
Fars_Fatal_Inj_vs_No_Inj	96.69	94.75	97.06	95.01
Average	88.39	87.11	88.52	87.19

Datasets	32 mappers			
	Chi-BigData-Max		Chi-BigData-Ave	
	Acc_{tr}	Acc_{tst}	Acc_{tr}	Acc_{tst}
RLCP	99.63	99.63	99.63	99.63
Kddcup_DOS_vs_normal	99.92	99.92	99.92	99.92
Poker_0_vs_1	61.27	58.93	61.82	59.30
Covtype_2_vs_1	74.69	74.62	74.88	74.85
Census	97.11	93.48	97.12	93.32
Fars_Fatal_Inj_vs_No_Inj	96.49	94.26	96.87	94.63
Average	88.19	86.81	88.37	86.94

Datasets	64 mappers			
	Chi-BigData-Max		Chi-BigData-Ave	
	Acc_{tr}	Acc_{tst}	Acc_{tr}	Acc_{tst}
RLCP	99.63	99.63	99.63	99.63
Kddcup_DOS_vs_normal	99.92	99.92	99.93	99.93
Poker_0_vs_1	60.45	57.95	60.88	58.12
Covtype_2_vs_1	74.67	74.52	75.05	74.96
Census	97.07	93.30	97.13	93.11
Fars_Fatal_Inj_vs_No_Inj	96.27	93.98	96.76	94.56
Average	88.00	86.55	88.23	86.72

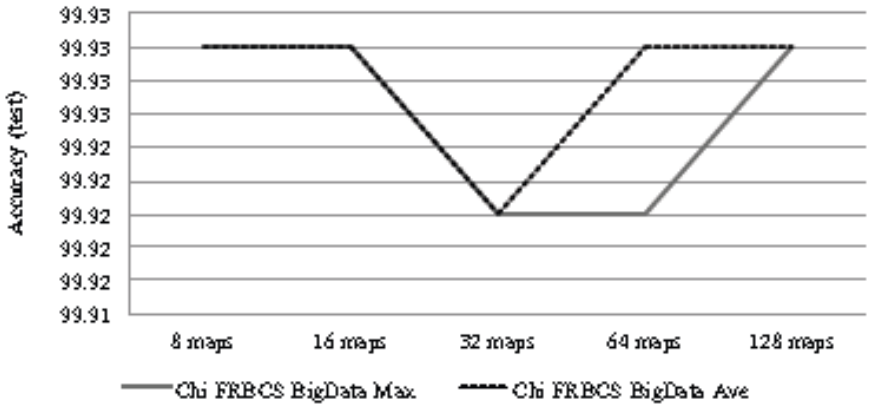
- **Performance improves slightly with less maps** (alleviate the small sample size problem)
- Chi-BigData-Ave obtains slightly better classification results

Big Data Classification with Fuzzy Models

Analysis of the Performance, Number of rules

Kddcup_DOS_vs_normal dataset	
NumRules by map	Final numRules
RB_1 size: 211	RB_R size: 301
RB_2 size: 212	
RB_3 size: 221	
RB_4 size: 216	
RB_5 size: 213	
RB_6 size: 210	
RB_7 size: 211	
RB_8 size: 214	

Robustness to the lack of data, increasing the final number of rules



(b) Kddcup_DOS_vs_normal dataset

Class	Instance Number
normal	972.781
DOS	3.883.370

Big Data Classification with Fuzzy Models

Analysis of the Performance, Number of rules

Datasets		8 maps	
	Chi-FRBCS	Chi-BigData-Max	Chi-BigData-Ave
	Average NumRules	Average NumRules	Average NumRules
Census	**31518.3**	34278.0	34278.0
Covtype_2_vs_1	**6962.7**	7079.1	7079.1
Fars_Fatal_Inj_vs_No_Inj	**16843.3**	17114.9	17114.9
Poker_0_vs_1	**51265.4**	52798.1	52798.1

Robustness to the lack of data, increasing the final number of rules

This may cause an improvement in the performance!!

Big Data Classification with Fuzzy Models

Analysis of the Performance, Runtime

Datasets	8 maps		
	Chi-FRBCS	Chi-BigData-Max	Chi-BigData-Ave
	Runtime (s)	Runtime (s)	Runtime (s)
Census	38655.60	**1102.45**	1343.92
Covtype_2_vs_1	86247.70	**2482.09**	2512.16
Fars_Fatal_Inj_vs_No_Inj	8056.60	**241.96**	311.95
Poker_0_vs_1	114355.80	**5672.80**	7682.19
Average	61828.93	**2374.82**	2962.56

KddCUP'99	
Class	Instance Number
normal	**972.781**
DOS	**3.883.370**

Maps number	Seconds
8	116.218,26
16	29.820,01
32	7.708,96
64	2.096,34
132	1.579,77

Performance improves slightly with less maps (alleviate the small sample size problem)

Fuzzy-based models for Big Data learning

BIG DATA with Fuzzy Models

- OUTLINE
  - Data fragmentation and lack of data problem
  - Big Data classification with fuzzy models
  - Are FRBCSs robust with respect to the lack of data?
  - Conclusions and future challenges

FUZZ-IEEE 2017 NAPLES, ITALY

BIG DATA

Experimental Framework

Table 1. Summary of BigData classification problems.

Datasets	#Ex.	#Atts.	Selected classes	#Samples per class
Covtype_1_vs_2	495,173	54	(1; 2)	(211,705; 283,468)
Poker_0	1,025,009	10	(0; remainder)	(513,701; 511,308)
Susy	4,923,622	18	(0;1)	(2,711,811;2,211,811)

Table 2. Configuration parameters for Chi-FRBCS-BigData.

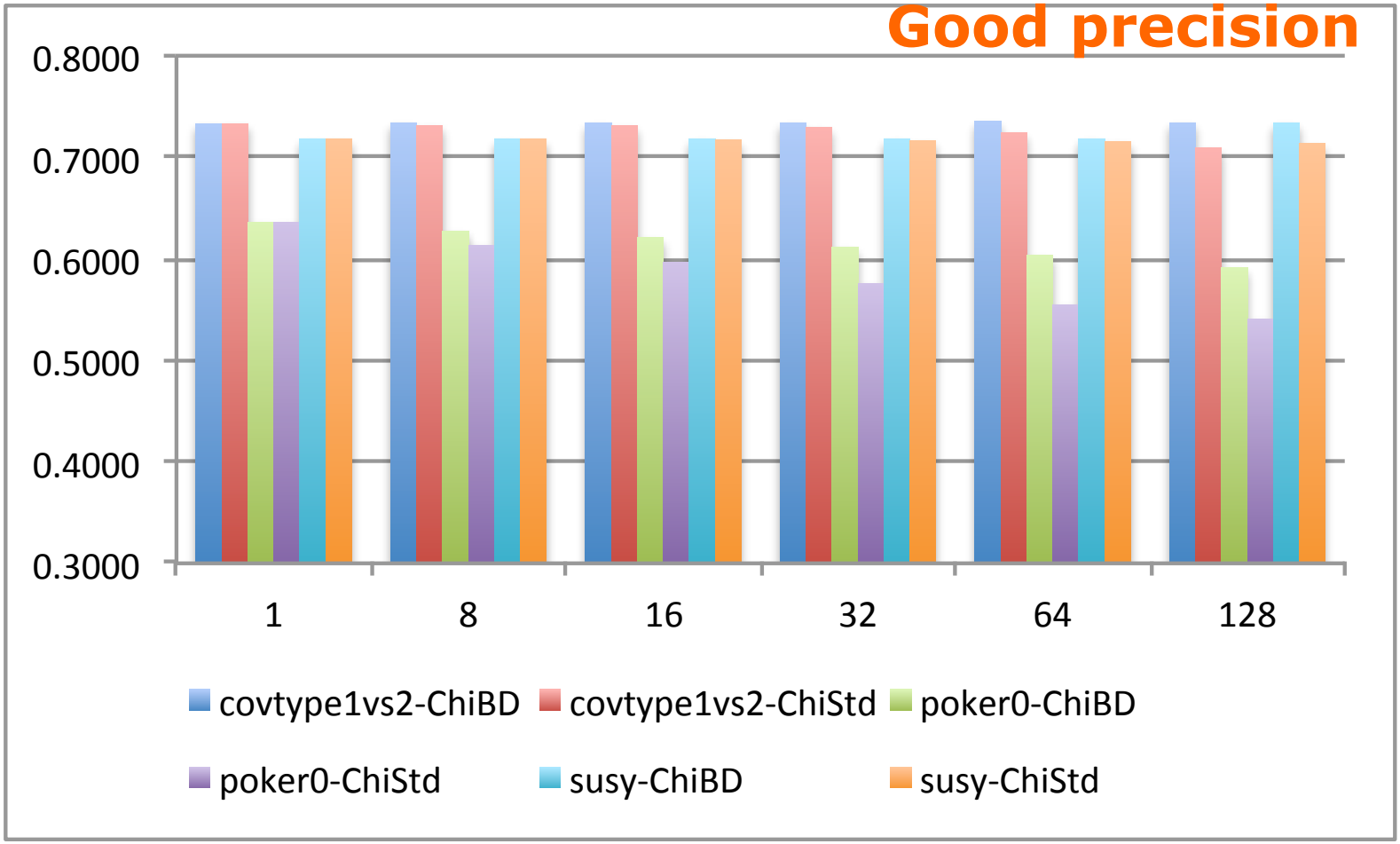
Number of Labels:	3 fuzzy partitions
Conjunction operator:	Product T-norm
Rule Weight:	Penalized Certainty Factor[7]
Fuzzy Reasoning Method:	Winning Rule

Two versions of Chi algorithm will be used:

- Chi-FRBCS-BigData (noted as **ChiBD**): MapReduce implementation
- Sequential Chi (noted as **ChiStd**): Results of 1Map (from 1 to 128)

Are FRBCS robust with respect to the lack of data?

BIG DATA with Fuzzy Models



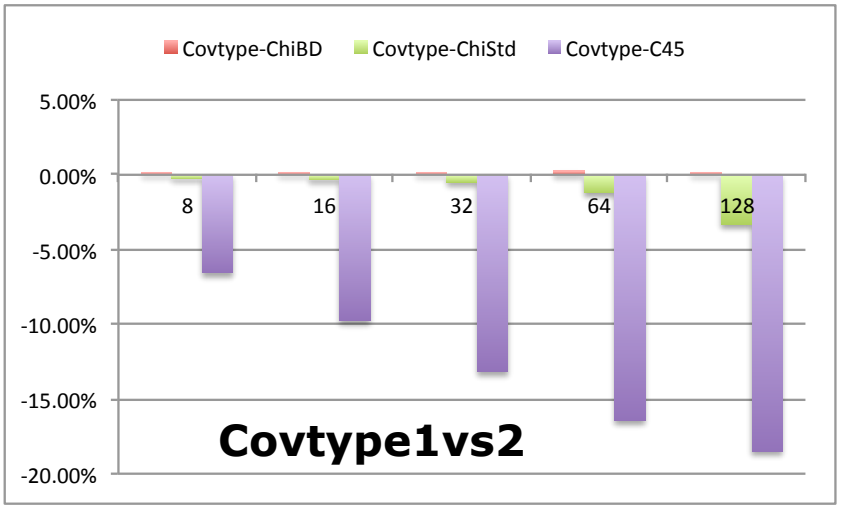
Accuracy obtained by Chi-FRBCSBigData (**ChiBD**) and a sequential run of Chi (**ChiStd**) with respect to **#Maps**

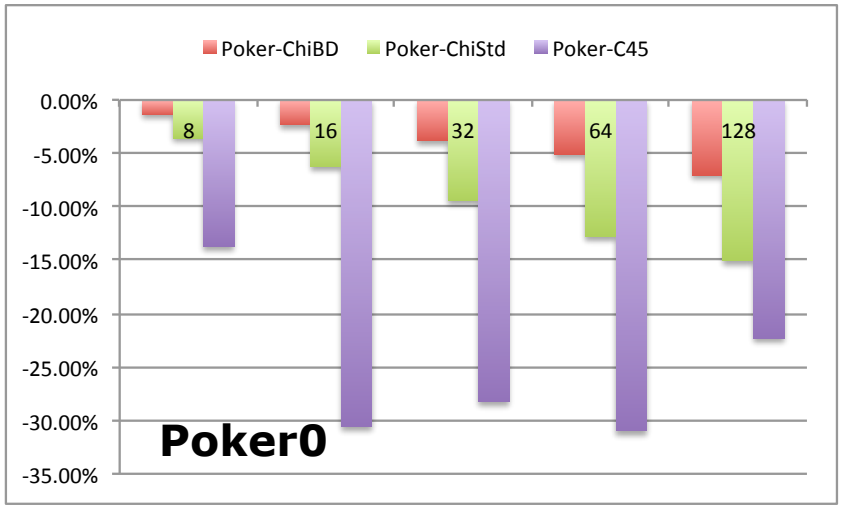
Are FRBCS robust with respect to the lack of data?

BIG DATA with Fuzzy Models

Analyzing the capabilities of representation of fuzzy models:

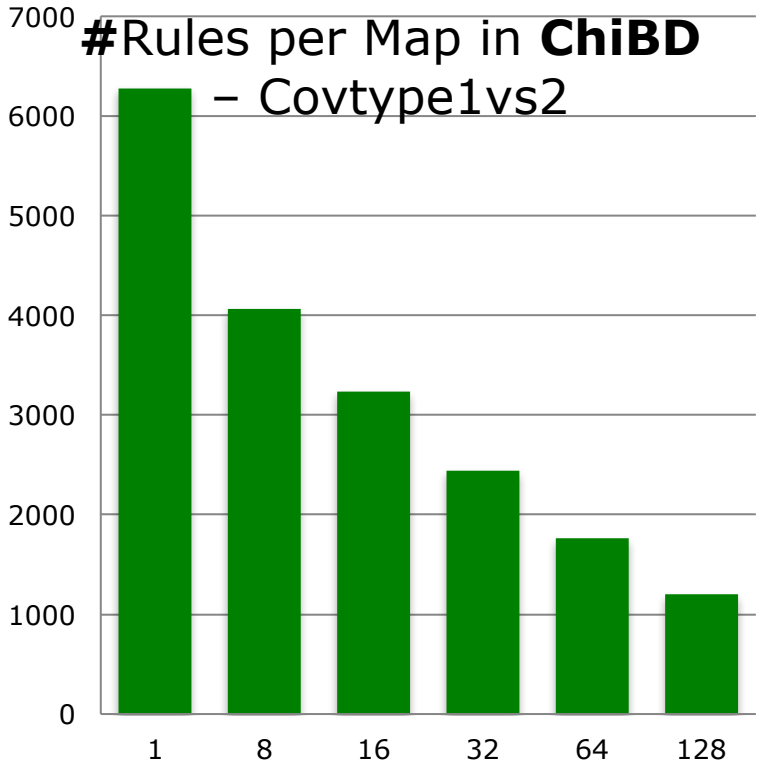
Variation of accuracy (%) 1 Map (100% data) from 8 to 128 Maps.
Algorithms: Chi-FRBCS-BigData (ChiBD), Chi (ChiStd) and C4.5

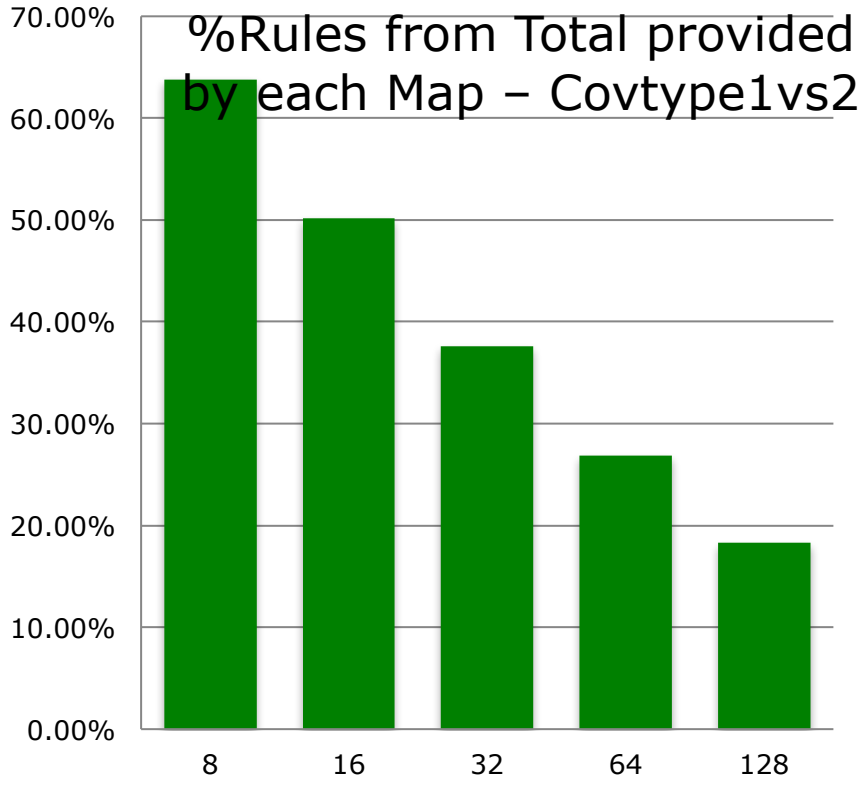




1. Advantage of aggregating the local models in MapReduce.
2. Fuzzy-Chi better than C4.5:
   - Few examples derive rules that represent larger regions.
   - How to provide a precise computation of the RWs.

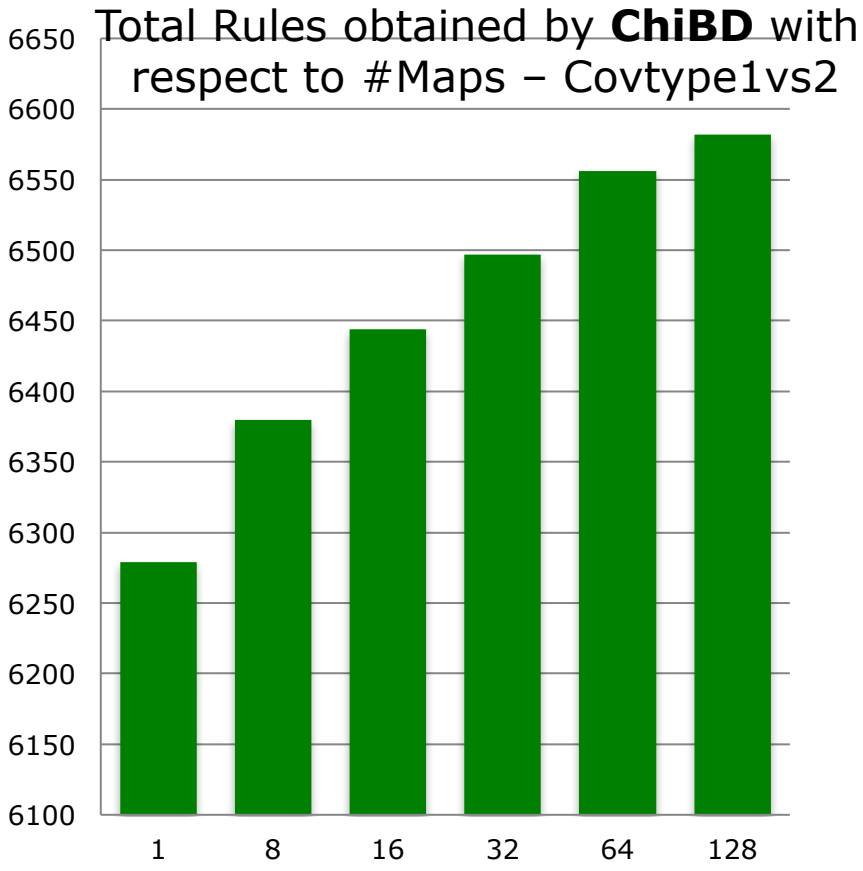
Analysis for the number of rules within Maps

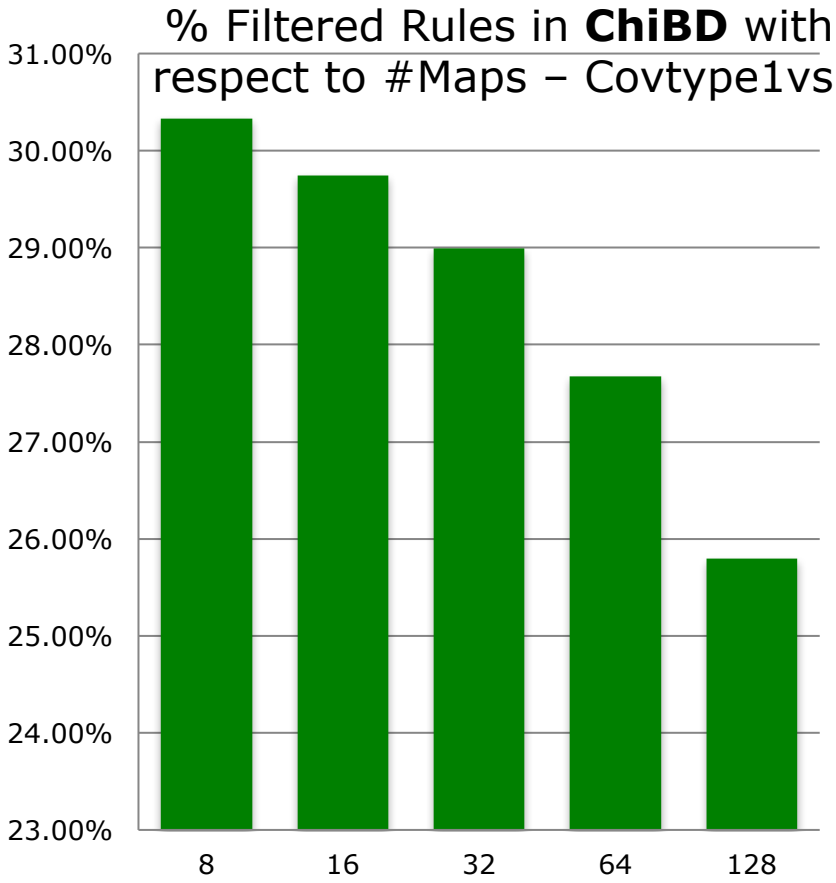




- Higher #maps, the smaller the contribution of each RBi.
- Better interpretability within each local fuzzy classifier,
- Few fuzzy rules: good representation of the problem space
- Global RB composed by more rules (RB_i aggregation)

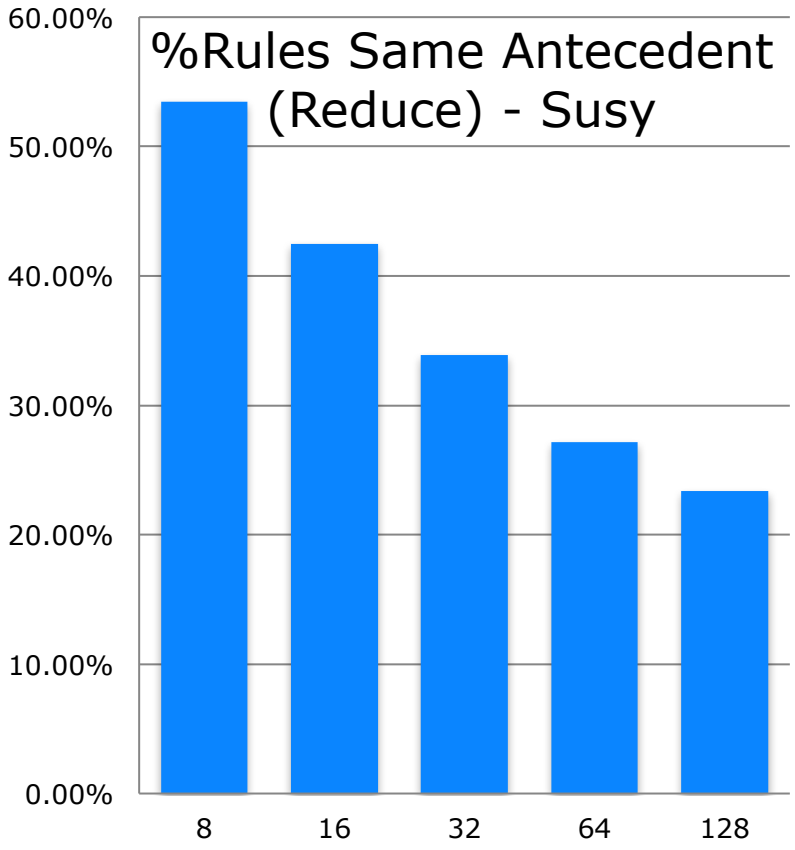
Analysis for the number of rules within Maps





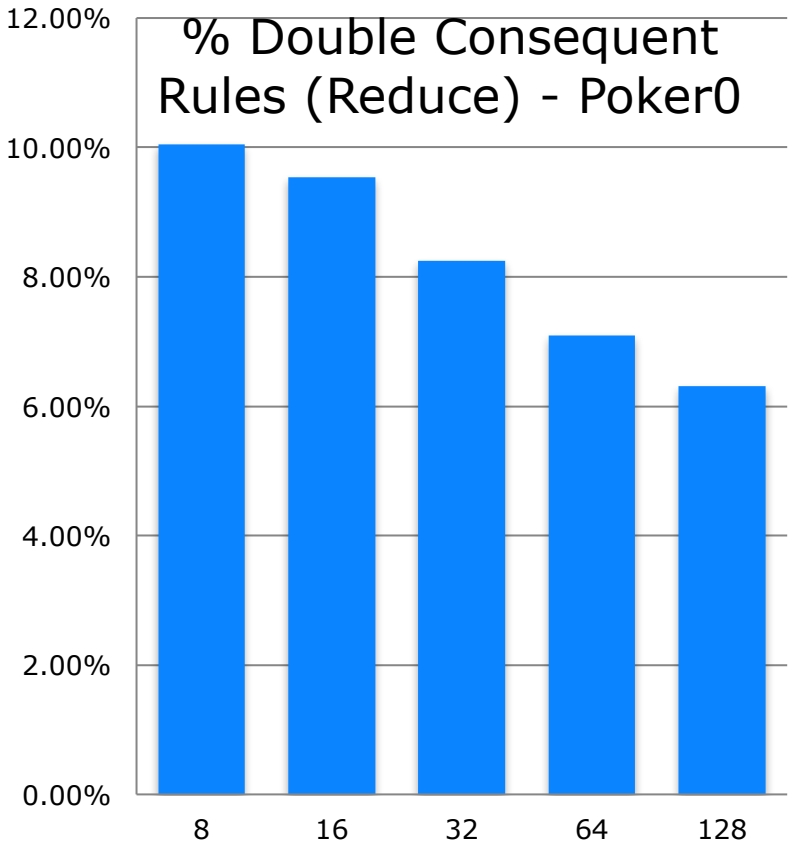
- **More Maps → More Rules.**
- P-CF RW: **Filter** rules (RW < 0) from local RBs.
- Sparsely distributed data: values of RWs become higher.

Fuzzy rules in Reduce: Same antecedent



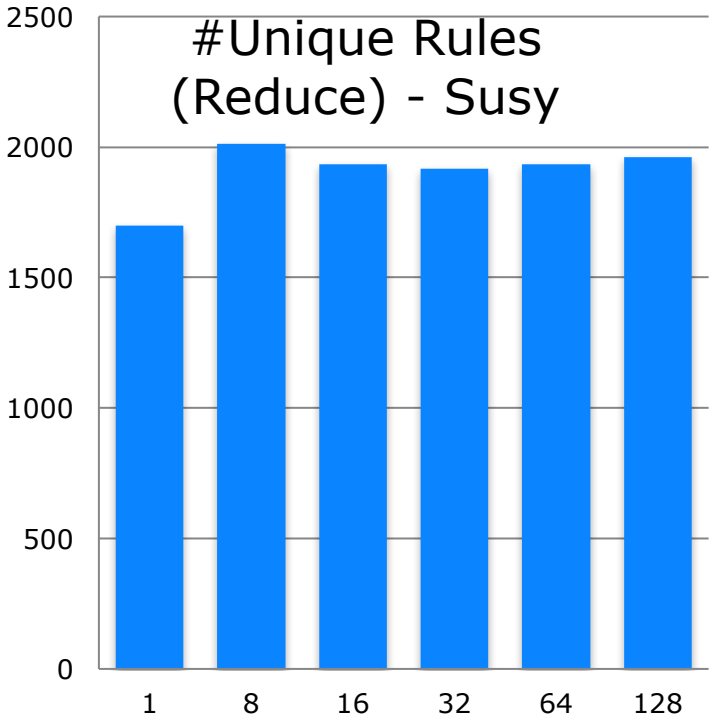
- High number of co-occurrences, especially for a low #maps:
  - The larger the volume of input data, more similar clusters are expected.
  - Same fuzzy labels chosen from these clusters
- Equal rules found among maps: higher density of data within the problem space.

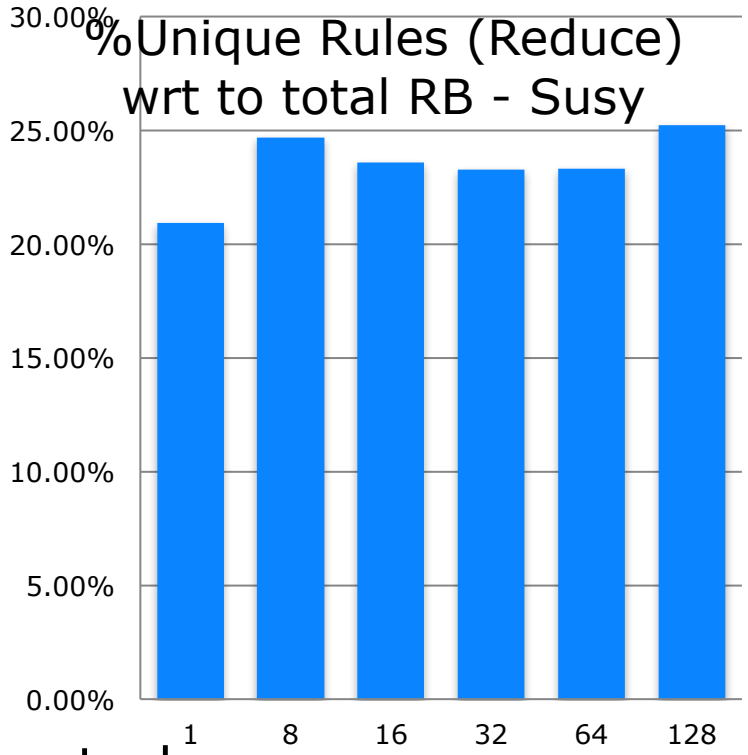
Fuzzy rules in Reduce: Double consequent



- How many rules in conflict arrive to a Reduce task.
- Rules related to overlapped regions.
- Low percentage:
  - Probably found within a single map process, and discarded due to the RW computation

Fuzzy rules in Reducer: Unique rules



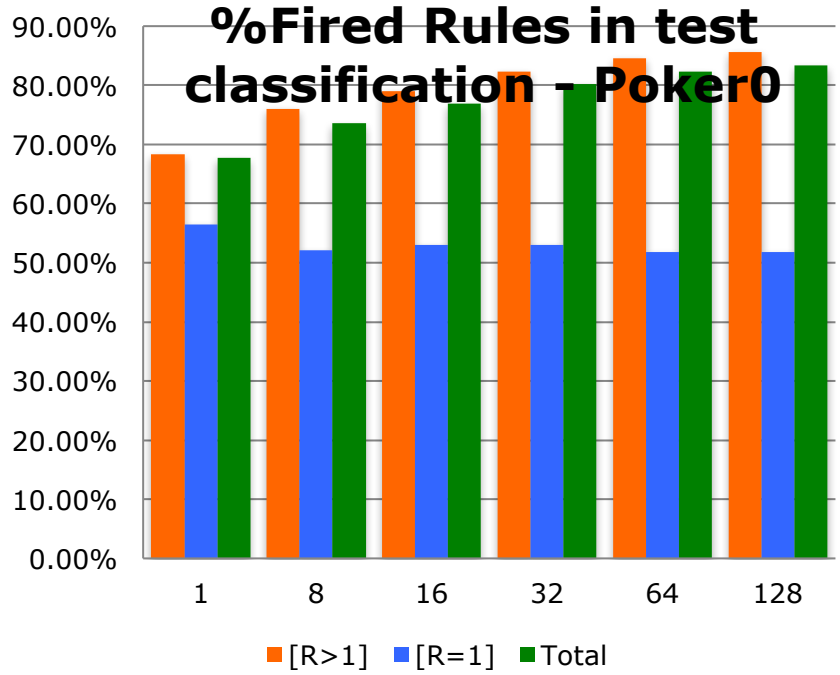


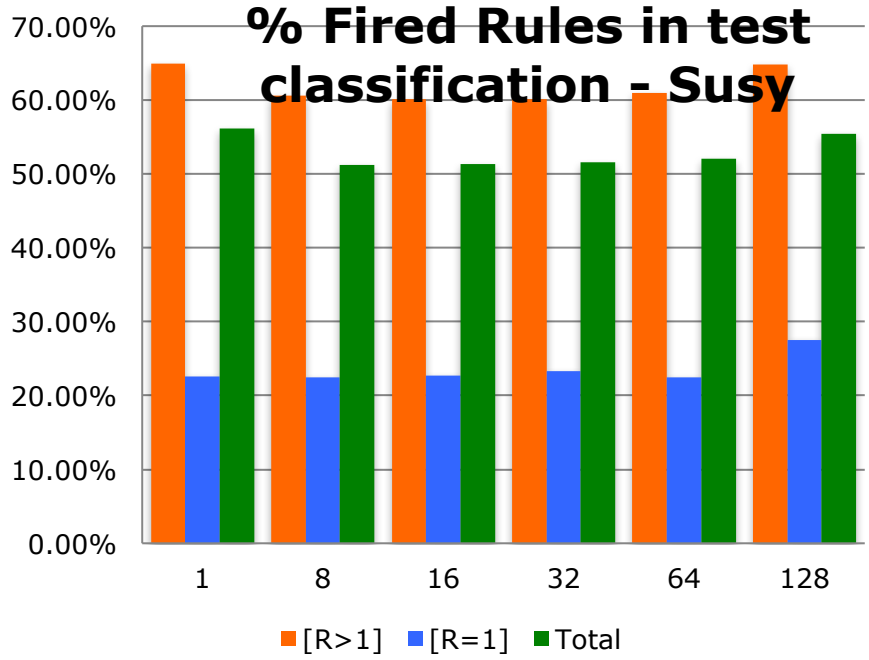
- Rules generated in a single Map task.
- This value is practically the same for all Map case studies.
- Instances far from areas of high density of data.
- Their influence is independent of the data distribution.

Contribution of fuzzy rules to classification

BIG DATA with Fuzzy Models

"Winning rule" inference: "strongest" rule determines class label





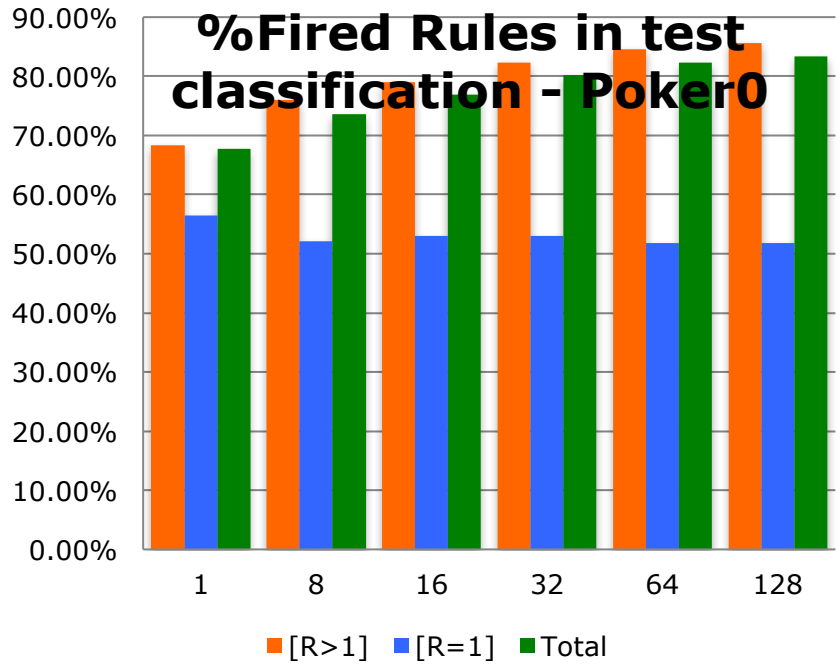
Types of rules analyzed:

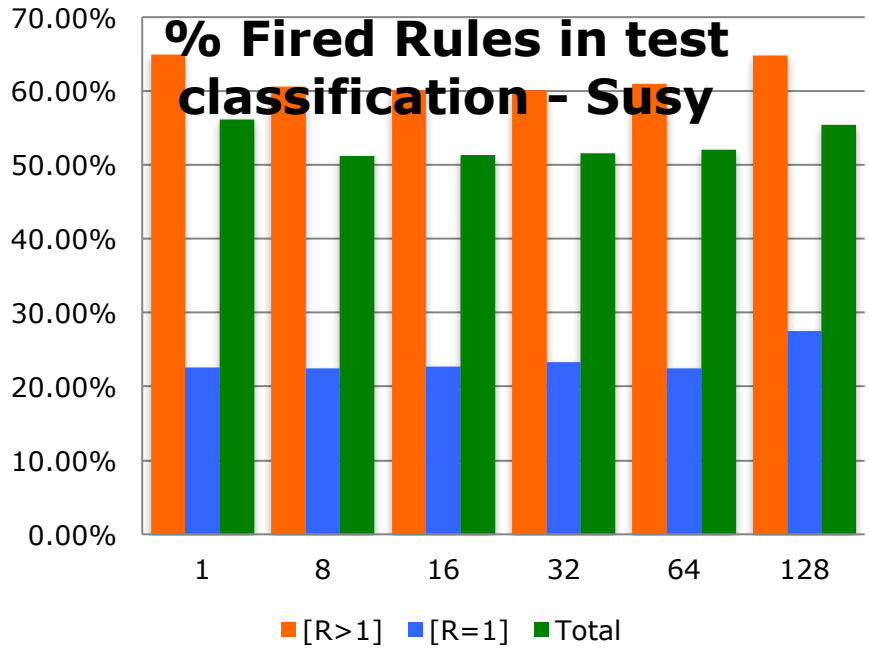
- All rules without distinction (noted as **"Total"**),
- Rules whose antecedents have been mined along different maps (noted as **[R>1]**), and
- Rules that are generated on a single map (**[R = 1]**).

Contribution of fuzzy rules to classification

BIG DATA with Fuzzy Models

"Winning rule" inference: "strongest" rule determines class label





- Not all rules are finally used for classification.
- Strong gap between "repeated" and "unique" rules
- The degree of contribution of "unique" rules is low.

Fuzzy-based models for Big Data learning

- OUTLINE
  - Data fragmentation and lack of data problem
  - Big Data classification with fuzzy models
  - Are FRBCSs robust with respect to the lack of data?
  - Conclusions and future challenges

Concluding Remarks

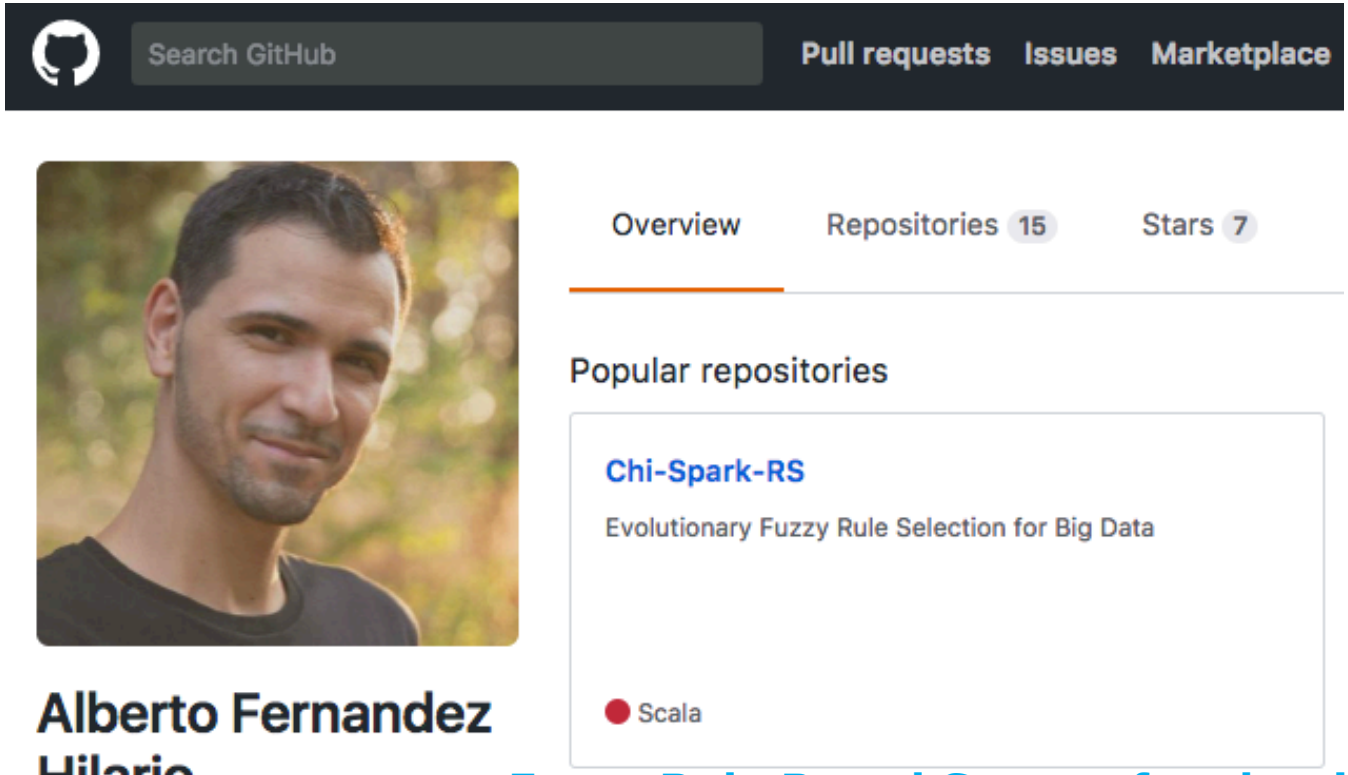
- **Linguistic fuzzy models for Big Data** under the MapReduce framework:
  - Manages big datasets
  - Without damaging the classification accuracy
  - Fast response times (increasing with the number of Maps)
- **Robustness** of fuzzy models when addressing the problem of **small disjuncts** and **lack of data**:
  - High degree of *replicated* rules along maps: partial RBs represent accurately a high percentage of the problem.
  - *Local models* learned in each Map task are of *high quality*. Aggregation boosts the recognition ability of the FRBCS.
  - Those fuzzy rules that are obtained from more than a map *contribute the most* to the final classification

Future Challenges

- Designing new learning methodologies for FRBCS in MapReduce:
  - Reduce phase for approximate fuzzy models
  - Deep analysis "ensembles vs. fusion of rules"
  - Analysis on small disjuncts preprocessing for fuzzy models
  - New fuzzy models based on accurate algorithms
  - Study the effect of *RW computation* in the quality of the fuzzy model
  - Application of good practices by means of scalable models:
    - Contextualization of the Data Base
    - Optimization of the Rule Base
- A promising line of work for the design of high performance Fuzzy Models for Big Data

Big Data Classification with Fuzzy Models

Code for our approaches: https://github.com/aFdezHilario



Fuzzy Rule Based System for classification (w/wo cost sensitive)

Evolutionary Fuzzy System for Rule Selection

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Smart Data

"Without Analytics, Big Data is just Noise"

F&D
FINANCE & DEVELOPMENT A QUARTERLY PUBLICATION OF THE INTERNATIONAL MONETARY FUND
September 2016 • Volume 53 • Number 3

The challenge is to translate raw data into information that can be used to improve performance.

Hal Varian
Google Chief Economist

***Guest post by Eric Schwartzman**, founder and CEO of Comply Socially*

http://www.briansolis.com/2013/04/without-analytics-big-data-is-just-noise/

Smart Data

Challenges of Big Data analysis

Jianqing Fan[1,*], Fang Han[2] and Han Liu[1]

National Science Review
1: 293–314, 2014
doi: 10.1093/nsr/nwt032
Advance access publication 6 February 2014

What are the challenges of analyzing Big Data?

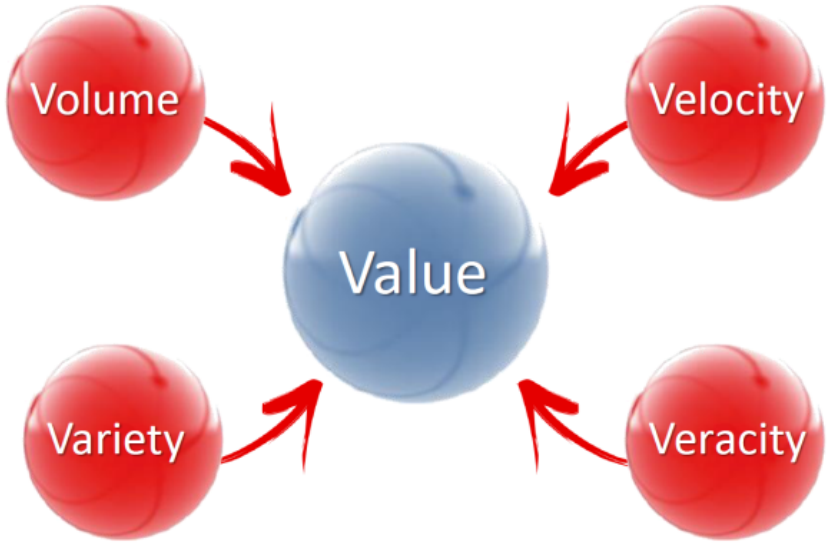
Big data are characterized by high dimensionality and large sample size. These two features raise three unique challenges: (i) high dimensionality brings noise accumulation, spurious correlations and incidental homogeneity, (ii) high dimensionality combined with large sample size creates l cost and algorithmic inst Big Data are typically aggreg ent time points using different technologies. This creates issues of heterogeneity, ical biases, and requires u ust procedures.

Noise accumulation

Spurious correlation

Smart Data

- RECALL: Big data as a concept is defined around five aspects:
- Data volume,
- Data velocity,
- Data variety,
- Data veracity and
- Data value.



Smart Data

a) While the **volume, variety and velocity** aspects refer to data generation process and how to capture and store the data,

b) Veracity and value aspects deal with the **quality and the usefulness of the data** leading to the point.

Smart Data

Smart Data (veracity and value) aims to filter out the noise and hold the valuable data, which can be effectively used by enterprises and governments for planning, operation, monitoring, control, and intelligent decision making.

What makes data smart? Three key attributes for data to be smart, it must be **accurate**, **actionable**, and **agile**.

Big Data (Analytics) → Smart Data

"Without Analytics, Big Data is just Noise"
Eric Schwartzman

The key is to explore how Big Data can become Smart Data.

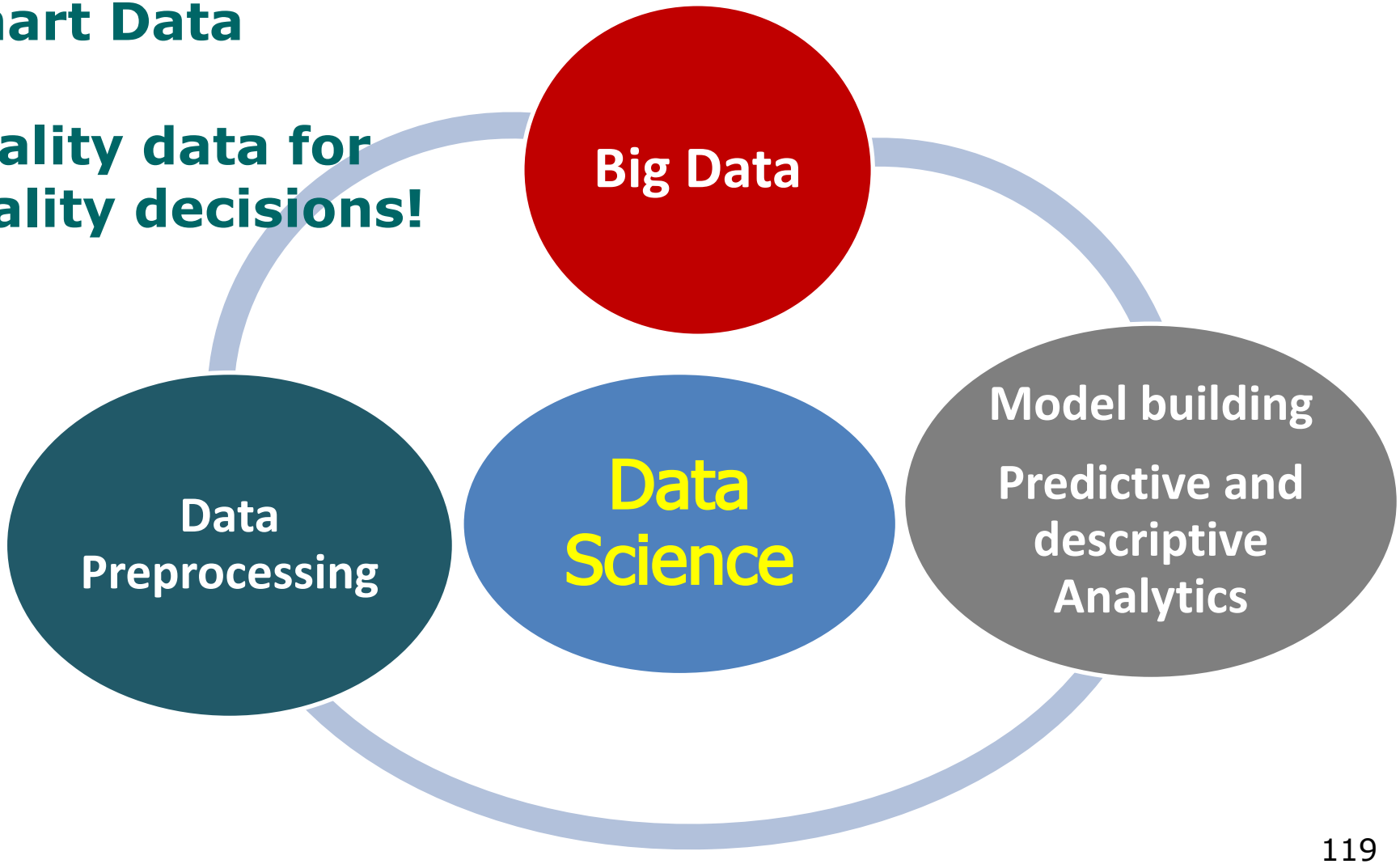
BIG DATA Preprocessing

Big Data
\+ Analytics
= Smart Data

Big Data (Analytics) → Smart Data

Smart Data

Quality data for quality decisions!



Big Data (Analytics) → Smart Data

What is included in data preprocessing?

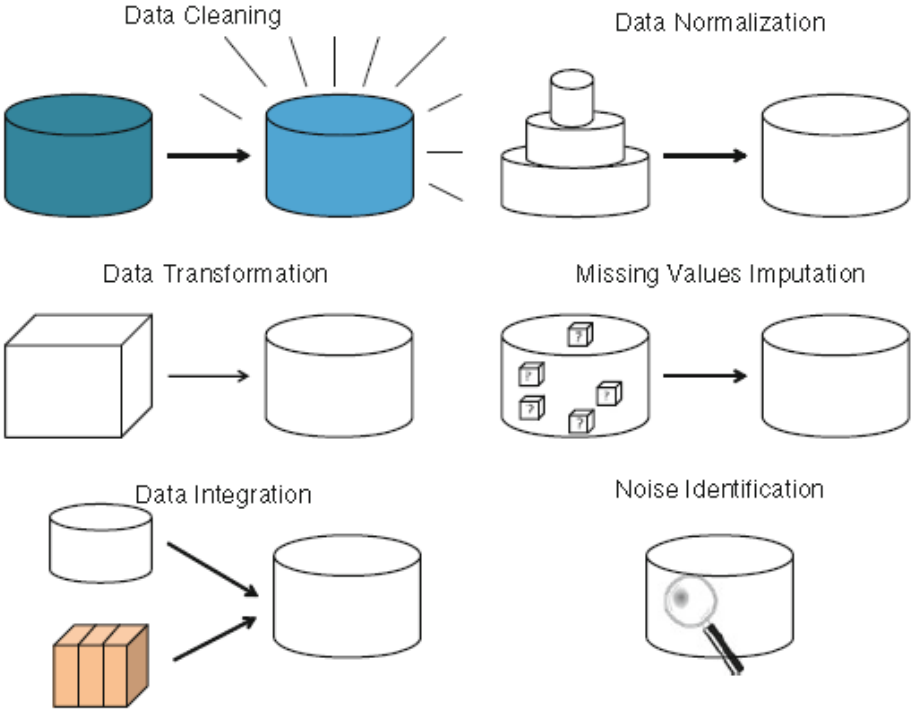


Fig. 1.3 Forms of data preparation

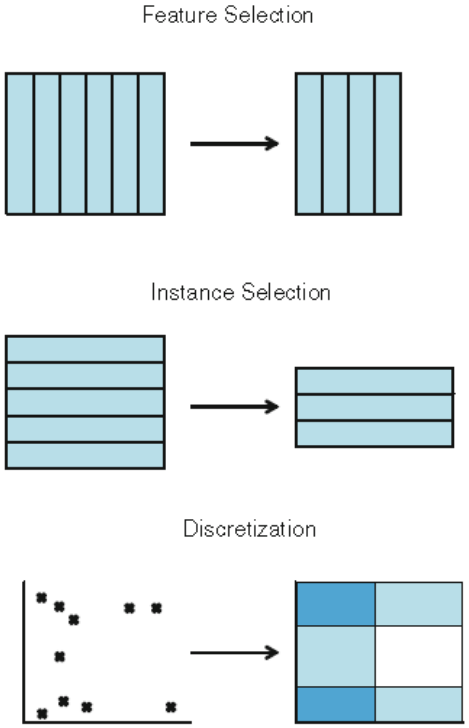


Fig. 1.4 Forms of data reduction

S. García, J. Luengo, F. Herrera
Data Preprocessing in Data Mining, Springer, 2015

Big Data Preprocessing → Smart Data

Big Data → Smart Data

Quality models based on quality data!

Big data preprocessing also must spend a very important part of the total time in a big data analytic process.

Credit: http://www.ibmsystemsmag.com/aix/storage/servers/The-Data-Squeeze/

Big Data at SCI²S - UGR

Bird's eye view SCI²S website

http://sci2s.ugr.es/BigData

Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

The web is organized according to the following summary:

1. Introduction to Big Data
2. Big Data Technologies: Hadoop ecosystem and Spark
3. Big Data preprocessing
4. Imbalanced Big Data classification
5. Big Data classification with fuzzy models
6. Classification Algorithms: k-NN
7. Big Data Applications
8. Dataset Repository
9. Literature review: surveys and overviews
10. Keynote slides
11. Links of interest

Big Data Preprocessing

1. Introduction to Data Preprocessing
2. Feature Selection
3. Feature Weighting
4. Discretization
5. Prototype Generation

Smart Data: Big Data Preprocessing

Bird's eye view SCI²S website

Feature Selection

http://sci2s.ugr.es/BigData

Fast-mRMR: an optimal implementation of minimum Redundancy Maximum Relevance algorithm

https://github.com/sramirez/fast-mRMR

This is big data based implementation of the classical feature selection method: minimum Redundancy and Maximum Relevance (mRMR);
(*Hanchuan Peng, Fuhui Long, and Chris Ding "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp.1226-1238, 2005*).

This includes several optimizations such as: cache marginal probabilities, accumulation of redundancy (greedy approach) and a data-access by columns.

Smart Data: Big Data Preprocessing

Feature Selection

http://sci2s.ugr.es/BigData

Fast-mRMR: Minimum Redundancy Maximum Relevance algorithm

Data Set	#Train Ex.	#Test Ex.	#Atts.	#Cl.	Sparse
epsilon	400 000	100 000	2000	2	No
dna	79 739 293	10 000 000	200	2	No
ECBDL14	65 003 913	2 897 917	630	2	No
url	1 916 904	479 226	3 231 961	2	Yes
kddb	19 264 097	748 401	29 890 095	2	Yes

TABLE IV: Selection time by dataset and threshold (in seconds)

# Features	kddb	url	dna	ECBDL14	epsilon
10	283.61	94.06	97.83	332.90	111.42
25	774.43	186.22	148.78	596.31	173.39
50	1365.82	333.70	411.84	1084.58	292.07
100	2789.55	660.48	828.35	2420.94	542.05

Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data.
S Ramírez-Gallego, I Lastra, D Martínez-Rego, V Bolón-Canedo, J.M. Benítez, F. Herrera, A. Alonso-Betanzos
International Journal of Intelligent Systems, Volume 32, Issue 2 February 2017 Pages 134–152

Smart Data: Big Data Preprocessing

TABLE IV: Selection time by dataset and threshold (in seconds)

# Features	kddb	url	dna	ECBDL14	epsilon
10	283.61	94.06	97.83	332.90	111.42
25	774.43	186.22	148.78	596.31	173.39
50	1365.82	333.70	411.84	1084.58	292.07
100	2789.55	660.48	828.35	2420.94	542.05

Data Set	#Atts.
epsilon	2000
dna	200
ECBDL14	630
url	3 231 961
kddb	29 890 095

From high dimensionality to small dimensionality:

Eases the application of fuzzy learning systems

Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data.
S Ramírez-Gallego, I Lastra, D Martínez-Rego, V Bolón-Canedo, J.M. Benítez, F. Herrera, A. Alonso-Betanzos
International Journal of Intelligent Systems, Volume 32, Issue 2 February 2017 Pages 134–152

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Beyond interpretability vs. accuracy trade-off

- Fuzzy systems began to be used for expert knowledge representation, knowledge linguistically expressed that allowed the development of **fuzzy logic controllers**.

1975-1995

- Later they were used **to represent knowledge extracted database** and began the rise of automatic learning of fuzzy systems, being developed multiple proposals with different tools such as neural networks, genetic algorithms, clustering, ad-hoc models, ...

1992-2016

- Then **we witness the long and rich debate about the balance between interpretability and precision fuzzy modeling**, and interpretability role played in the design and subsequent use of fuzzy systems.

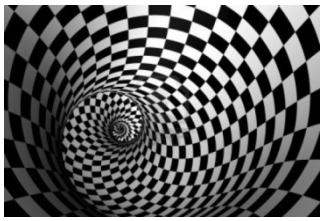
1992-2016

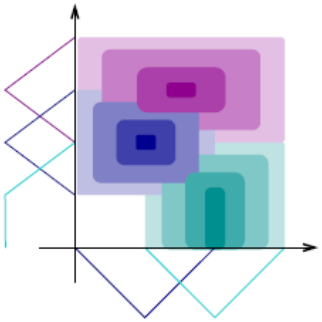
Beyond interpretability vs. accuracy trade-off

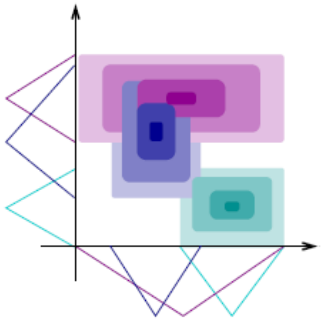
Learning from data **25 years studies 1992-2016**

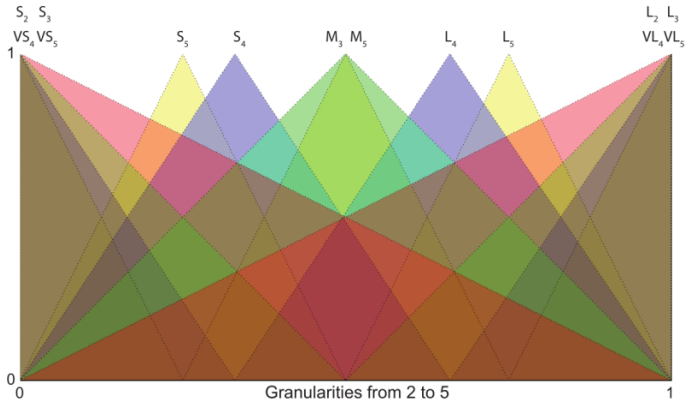
We are caught in a spiral

Interpretability vs Precision









Debate as a dialectical exercise that lasts and that in most cases does not end with conclusions that have real value at a practical level.

The horizon is lost and we have not progressed toward open problems.

Beyond interpretability vs. accuracy trade-off

Several questions can be raised to reflect on the interpretability:

- When is useful the linguistic description or the fuzzy rules interpreted with α-cuts?
- Why must we bet on the use of models of fuzzy rules models such as divide and conquer decision trees or crisp rule based systems?
- Are useful interpretable models for complex problems with lots of data?
- Should we continue to design more complex and not interpretable fuzzy models via models fusion as ensembles, to be competitive in precision?

Beyond interpretability vs. accuracy trade-off

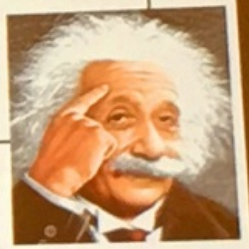
Suggestion to design "fuzzy intelligent systems" from data

N. Pal
WCCI2016

Designing "Intelligent Systems" from data

What kind of systems do we want?

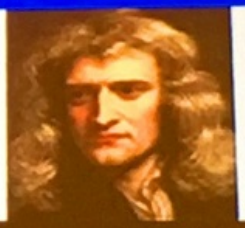
- They should perform well
- They should be understandable (not a black box)
- They should be simple (why!!)



"Make everything as simple as possible, but not simpler."

"I am not a genius, I am just curious. I ask many questions, and when the answer is simple, then God is answering." (Albert Einstein)

"Nature is pleased with simplicity. And nature is no dummy" (Isaac Newton)



performing well, understandable and simple

Beyond interpretability vs. accuracy trade-off

There are other models beyond rule-based models **(performing well, understandable and simple)**.

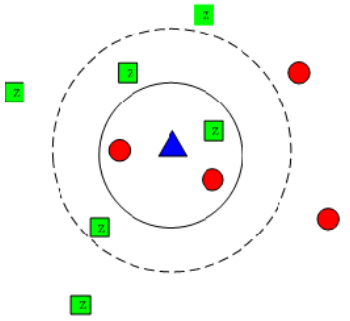
An interesting model: fuzzy prototypes represented by nearest neighbor techniques.

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS,

580 VOL. SMC-15, NO. 4, JULY/AUGUST 1985

A Fuzzy *K*-Nearest Neighbor Algorithm

JAMES M. KELLER, MICHAEL R. GRAY, AND JAMES A. GIVENS, JR.



$$u_i(x) = \frac{\sum_{j=1}^{K} u_{ij}\left(1/\|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^{K}\left(1/\|x - x_j\|^{2/(m-1)}\right)}.$$

It deserves to go back over this model and to extend his use in data science/big data.

Beyond interpretability vs. accuracy trade-off

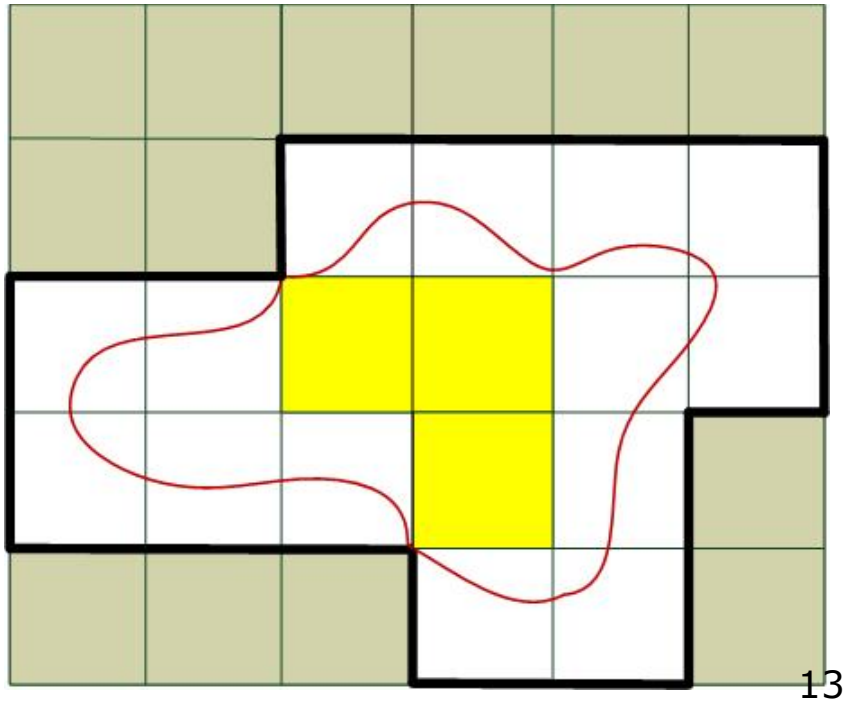
There are other models beyond rule-based models.

Using the similarity among data: Rough Sets Theory *(Lower approximation and positive region vs Upper approximation and negative region)*

$$Ry = \{x \mid x \in X \text{ and } (x, y) \in R\}$$

$$y \in R{\downarrow}A \quad \text{iff} \quad [y]_R \subseteq A$$
$$y \in R{\uparrow}A \quad \text{iff} \quad [y]_R \cap A \neq \emptyset$$

$$y \in R{\downarrow}A \quad \text{iff} \quad (\forall x \in X)((x, y) \in R \Rightarrow x \in A)$$
$$y \in R{\uparrow}A \quad \text{iff} \quad (\exists x \in X)((x, y) \in R \wedge x \in A)$$



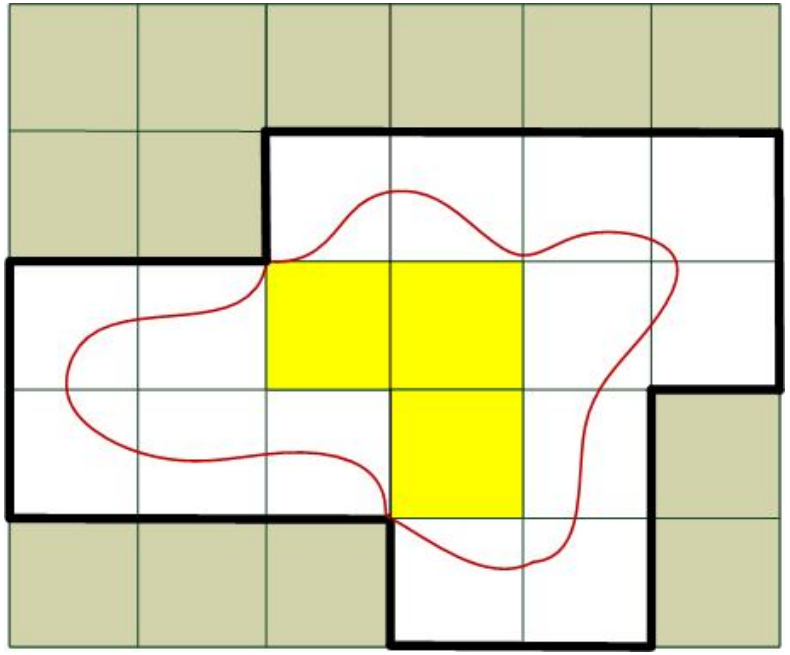
Beyond interpretability vs. accuracy trade-off

Using the similarity among data: Fuzzy Rough Sets Theory

$$Ry(x) = R(x, y)$$

$$R \downarrow A(y) = \inf_{x \in X} \mathcal{I}(R(x, y), A(x))$$

$$R \uparrow A(y) = \sup_{x \in X} \mathcal{T}(R(x, y), A(x))$$



State of the art

S Vluymans, L D'eer, Y Saeys, C Cornelis

Applications of Fuzzy Rough Set Theory in Machine Learning: a Survey

Fundamenta Informaticae 142 (1-4), 53-86, 2015

An application for imbalanced data sets – State of the art

IFROWANN: imbalanced fuzzy-rough ordered weighted average nearest neighbor classification

E Ramentol, S Vluymans, N Verbiest, Y Caballero, R Bello, C Cornelis, F. Herrera

IEEE Transactions on Fuzzy Systems 23 (5), 1622-1637, 2015.

Beyond interpretability vs. accuracy trade-off

They are additional fuzzy models in the literature, whose use is not analyzed at a theoretical or practical level in depth:

Fuzzy decision trees, fuzzy SVM, …

Today it is accepted that fuzzy systems are a good tool in data modeling, although not yet have developed studies analyzing the role "when and where they should be used and why"

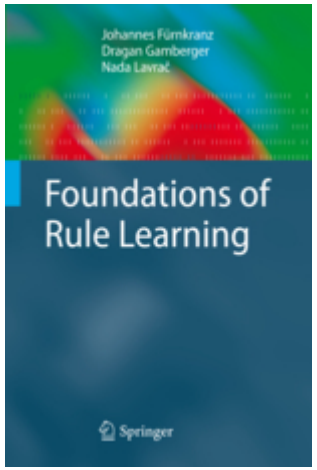
This is a challenge beyond interpretability-accuracy debate!!.

Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Fuzzy Big Data Science: Opportunities

- The most effective classification algorithms do not allow to interpret the models.
- The interpretation of the models is an essential feature of rule-based systems (RBSs)
- There are applied areas where experts want to know why decisions.
- RBSs fill a very important gap in Data Science.

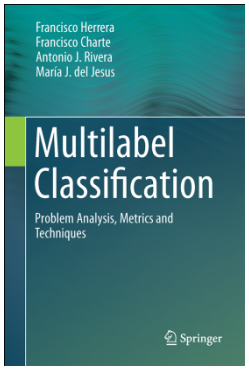


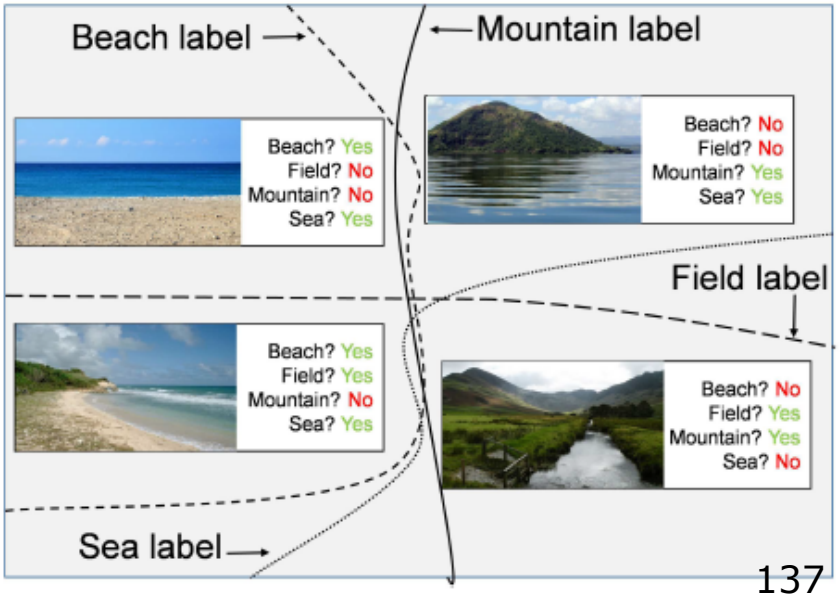
Fuzzy Rule Based Systems play an important role as RBSs for the description of the facts in terms of knowledge represented by rules.

Fuzzy Big Data Science: Opportunities

There are open multiple scenarios of data science and their challenges and opportunities for fuzzy modeling:

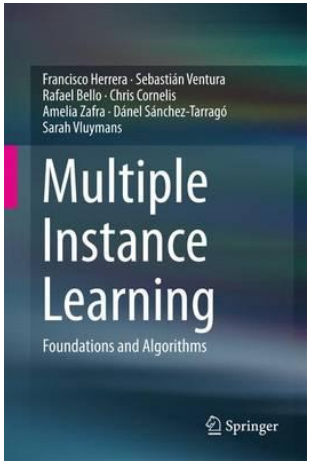
- Multi Label
- Multi-Instance Learning
- Multiview Lear
- Semi-supervised Le
- Transfer Learning
- Big Data





Fuzzy Big Data Science: Opportunities

There are open multiple scenarios of data science and their challenges and opportunities for fuzzy modeling:

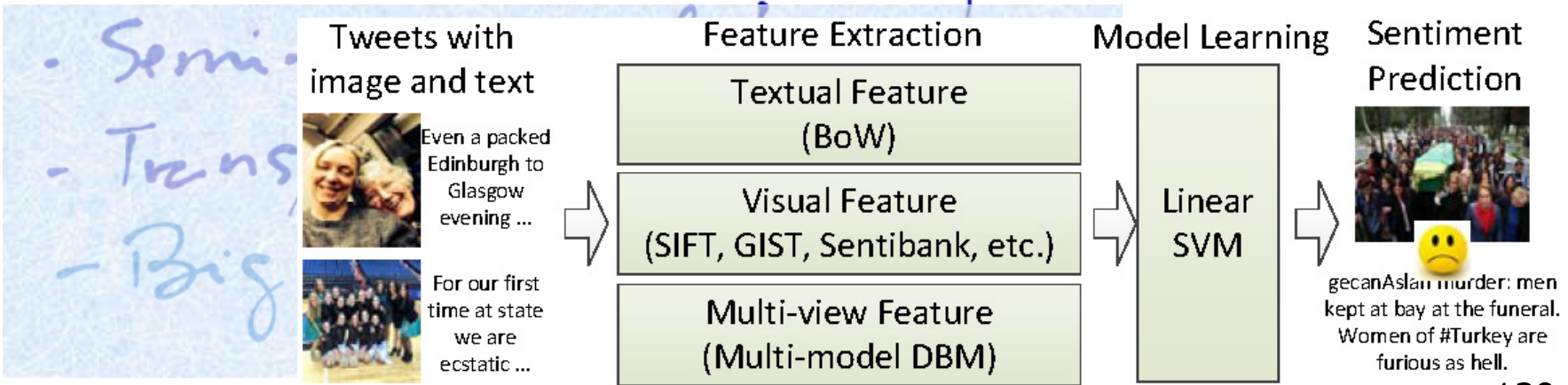


- Multi Label
- Multi. Instance Learning
- Multiview Learning
- Semi
- Tra
- Big Data

Fuzzy Big Data Science: Opportunities

There are open multiple scenarios of data science and their challenges and opportunities for fuzzy modeling:

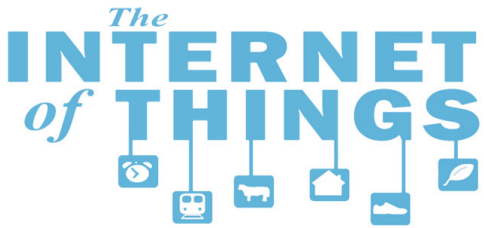
- Multi Label
- Multi-Instance Learning
- Multiview Learning

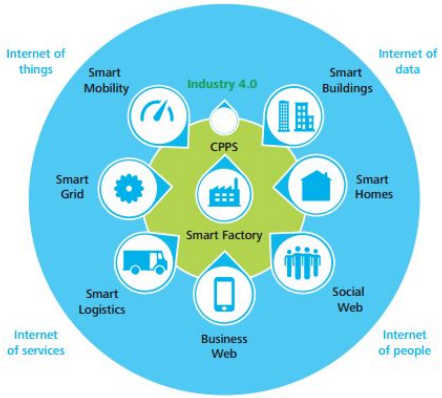


Fuzzy Big Data Science: Opportunities

There are open multiple scenarios of data science and their challenges and opportunities for fuzzy modeling:

- Multi Label
- Multi-Instance Learning
- Multiview Learning
- Semi-supervised Learning
- Transfer Learning
- Big Data





Outline

- Introduction to Big Data
- Big data analytics
- Fuzzy-based models for Big Data Learning
- Smart Data: The missing bridge between Big Data and real applications to get high quality data
- Fuzzy Data Science: Beyond Interpretability-Accuracy Tradeoff
- Fuzzy Big Data Science: Opportunities
- Final Comments

Final Comments

Data Mining, Machine learning and data preprocessing: Huge collection of algorithms

Big Data Analytics

Big Data: A small subset of algorithms

Big Data Preprocessing:
A few methods for preprocessing in Big Data analytics. Evolutionary models are a promising approach.

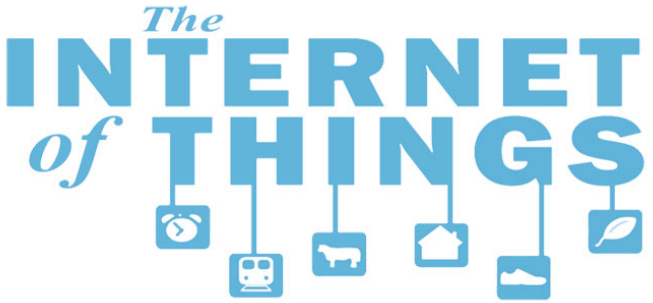
Soft computing approaches are useful to tackle big data analytics

Final Comments

- **Fuzzy models for big data: Robustness to the lack of data for the data fragmentation, producing high performance (accuracy).**
- **The focus should be on**
  - **a) designing models for maps and**
  - **b) the combination phase (reduce).**
- **Potential applications in multiple scenarios:**

BigData&
Analytics
and
Non Standard
Classification

Social Big Data



Final Comments

"We have to place the Fuzzy Systems on the stage where they can play an important role in Data Science".

Deep Learning: Transforms information to provide knowledge in terms of decisions, but do not explain why decisions (resurface neural networks).

Fuzzy Rule Based Systems/Fuzzy Models: Interpret information to provide knowledge in the form of linguistic rules or understandable and simple fuzzy models. They should have a greater presence in the Data Science forums **(performing well, understandable and simple)**.

Final Comments: Questions for discussion

- **Interpretabilty.** Are we caught in a spiral? Why more discussions on interpretability? What are the limits?
- **Beyond classical classification and regressions.** What are the possibilities of fuzzy systems in the new data science areas?
- **Big Data.** Why to design fuzzy systems for big data?
- **Deep learning in the center of the Artificial Intelligence**. **Deep Learning vs fuzzy systems.** Why to use fuzzy systems? What is the position of Fuzzy logic in the Artificial Intelligence board? "Designing Intelligent Fuzzy Systems performing well, understandable and simple"

Final Comments

Big data and analytics: a large challenge offering great opportunities

- **A small subset of algorithms.**
  It is necessary to redesign new algorithms
- **Computing Model**
  - **Accuracy and Approximation**
  - **Efficiency requirements for Algorithm**

- **Big Data Preprocessing**
- **Noise in data distorts**
  Need automatic methods for "cleaning" the data
- **Missing values management**
- **Big Data Reduction**

Quality data for quality models in big data analytics

Final Comments

Quality decisions must be based on quality big data and quality models.

Computational Intelligence may be very useful to create new high quality big data analytics models.

Big Data at SCI²S - UGR

Bird's eye view SCI²S website

http://sci2s.ugr.es/BigData

Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes

The web is organized according to the following summary:

1. Introduction to Big Data
2. Big Data Technologies: Hadoop ecosystem and Spark
3. Big Data preprocessing
4. Imbalanced Big Data classification
5. Big Data classification with fuzzy models
6. Classification Algorithms: k-NN
7. Big Data Applications
8. Dataset Repository
9. Literature review: surveys and overviews
10. Keynote slides
11. Links of interest

Fuzzy Models for Data Science and Big Data

Fuzzy Models for Data Science and Big Data

Francisco Herrera and Alberto Fernández

Soft Computing and Information Intelligent Systems (SCI²S)

University of Granada, Spain

Email: {herrera, alberto}@decsai.ugr.es

http://sci2s.ugr.es

Isaac Triguero

School of Computer Science

Automated Scheduling Optimisation and Planning (ASAP)

University of Nottingham

United Kingdom

Isaac.Triguero@nottingham.ac.uk

UNIVERSIDAD DE GRANADA

University of Nottingham

UK | CHINA | MALAYSIA