An exact scalable method of the k nearest neighbour algorithm to handle big data sets

BIG DATA II

Jesús Maillo (jesusmh@decsai.ugr.es)

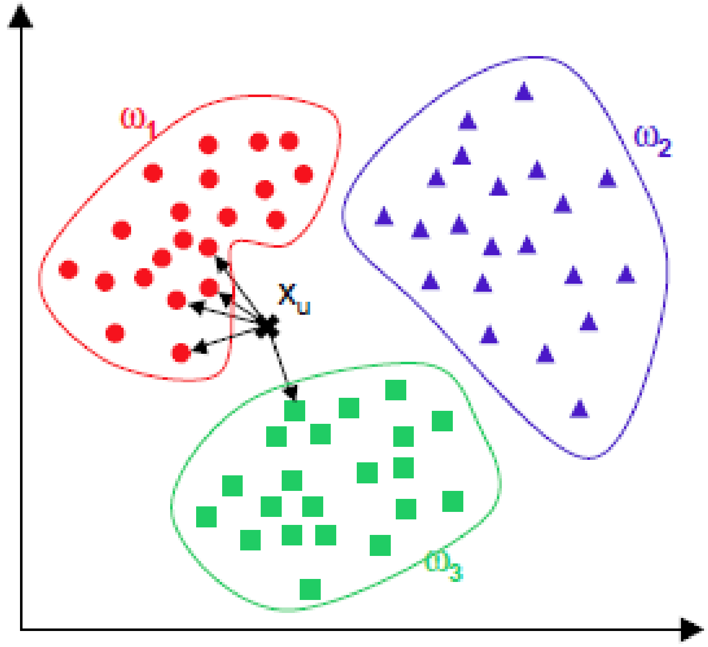
08/03/2018

Outline

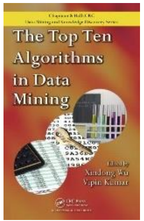
- **Introduction & Preliminaries**
- kNN in the big data
- kNN-IS: A MapReduce implementation for kNN under Apache Spark
- Fuzzy kNN for big data

KNN

Data mining: Classification



- Instance-based Learning - IBL
  - No “training phase”
- K Nearest Neighbours - kNN
  - It highlights because of its simplicity and effectiveness.

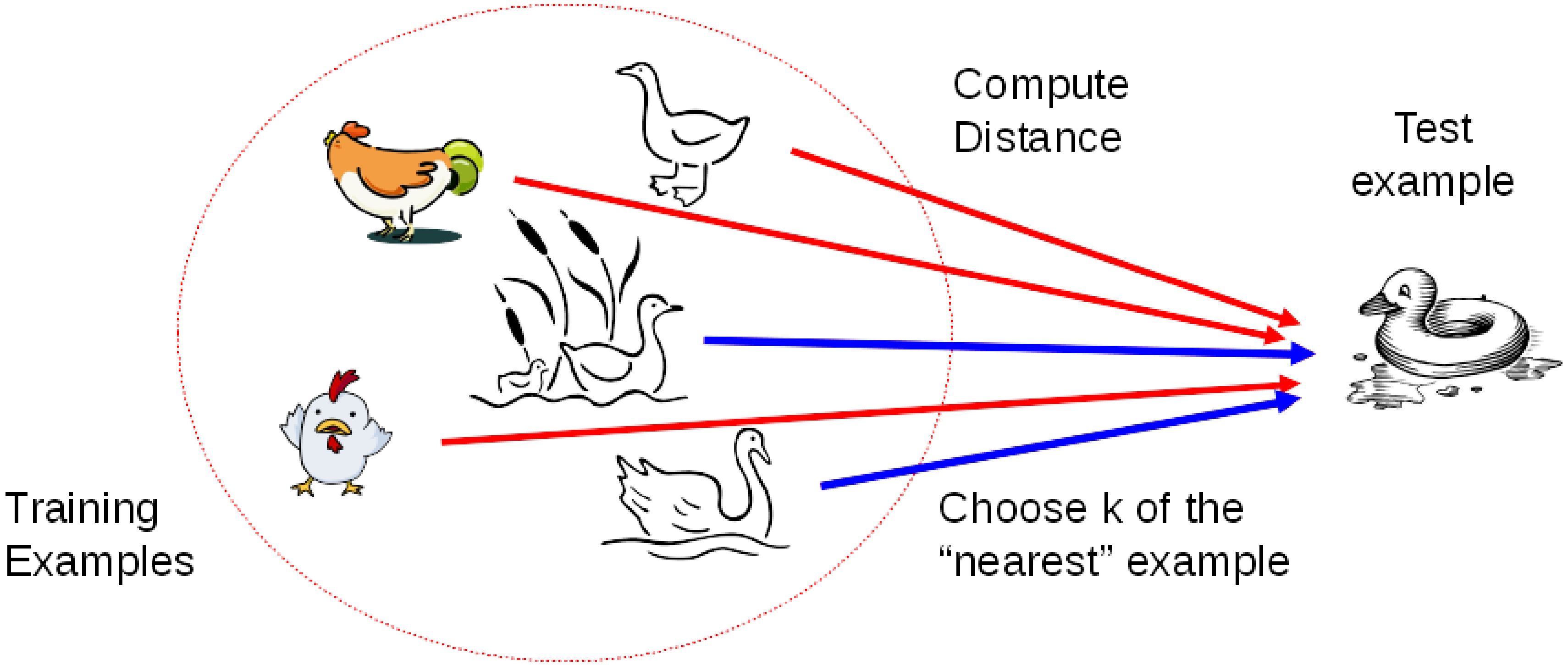


X. Wu and V. Kumar, **The Top Ten Algorithms in Data Mining,** Chapman & Hall/CRC Data Mining and Knowledge Discovery, 2009.

kNN: Basic idea

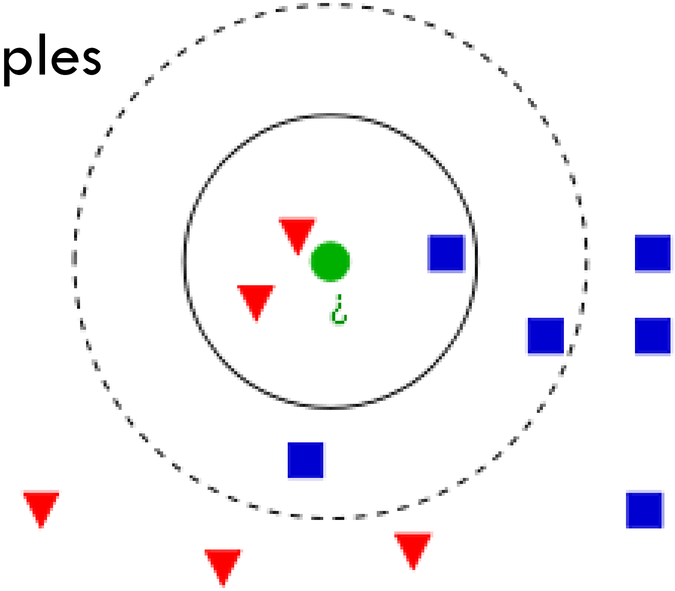
If it walks like a duck, quacks like a duck,

Then it's probably a duck



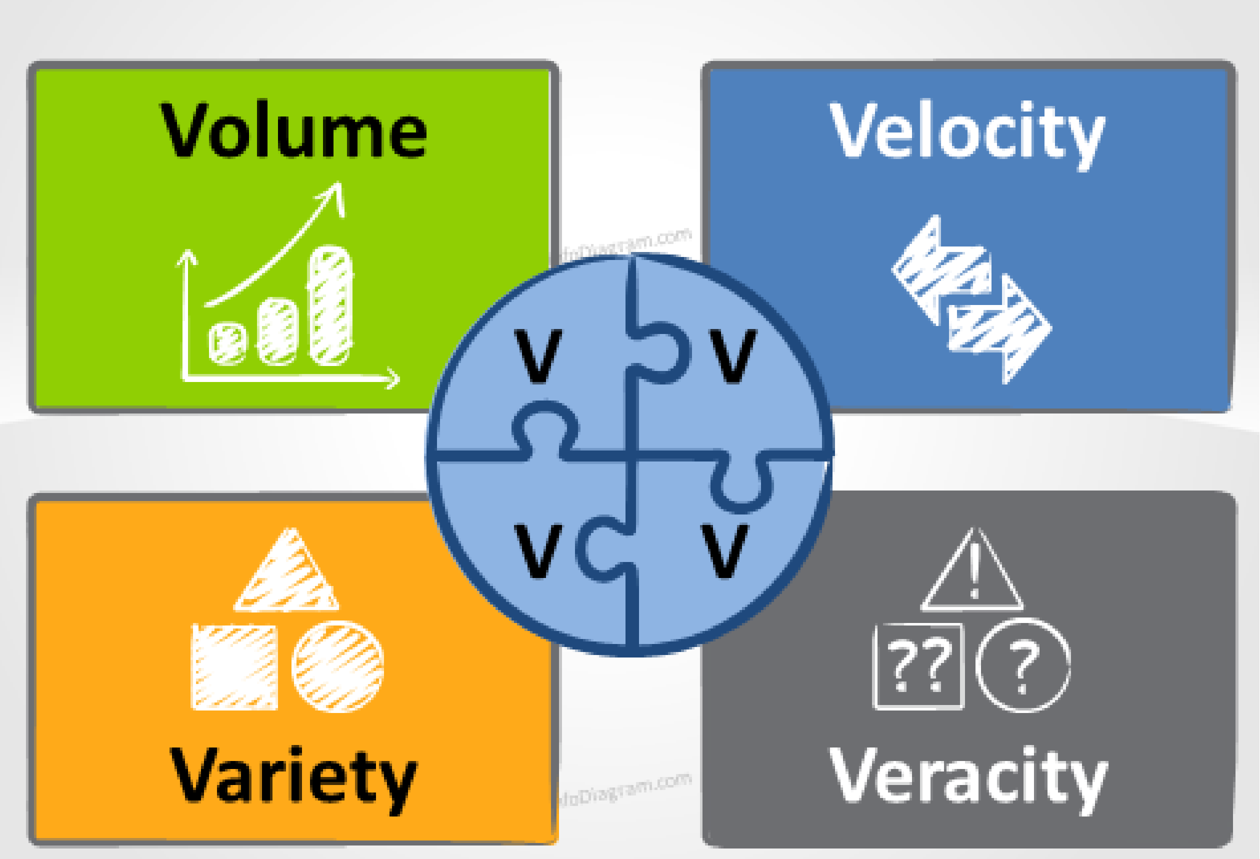
kNN: Basic idea

- Requires three things:
  - The set of training examples (TR)
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbours to consider
- To classify an unknown instance from a test set (TS):
  - Compute distance to other training examples
  - Identify the *k* nearest neighbours
  - Use class labels of nearest neighbours to determine the class label of unknown example (by taking majority vote)



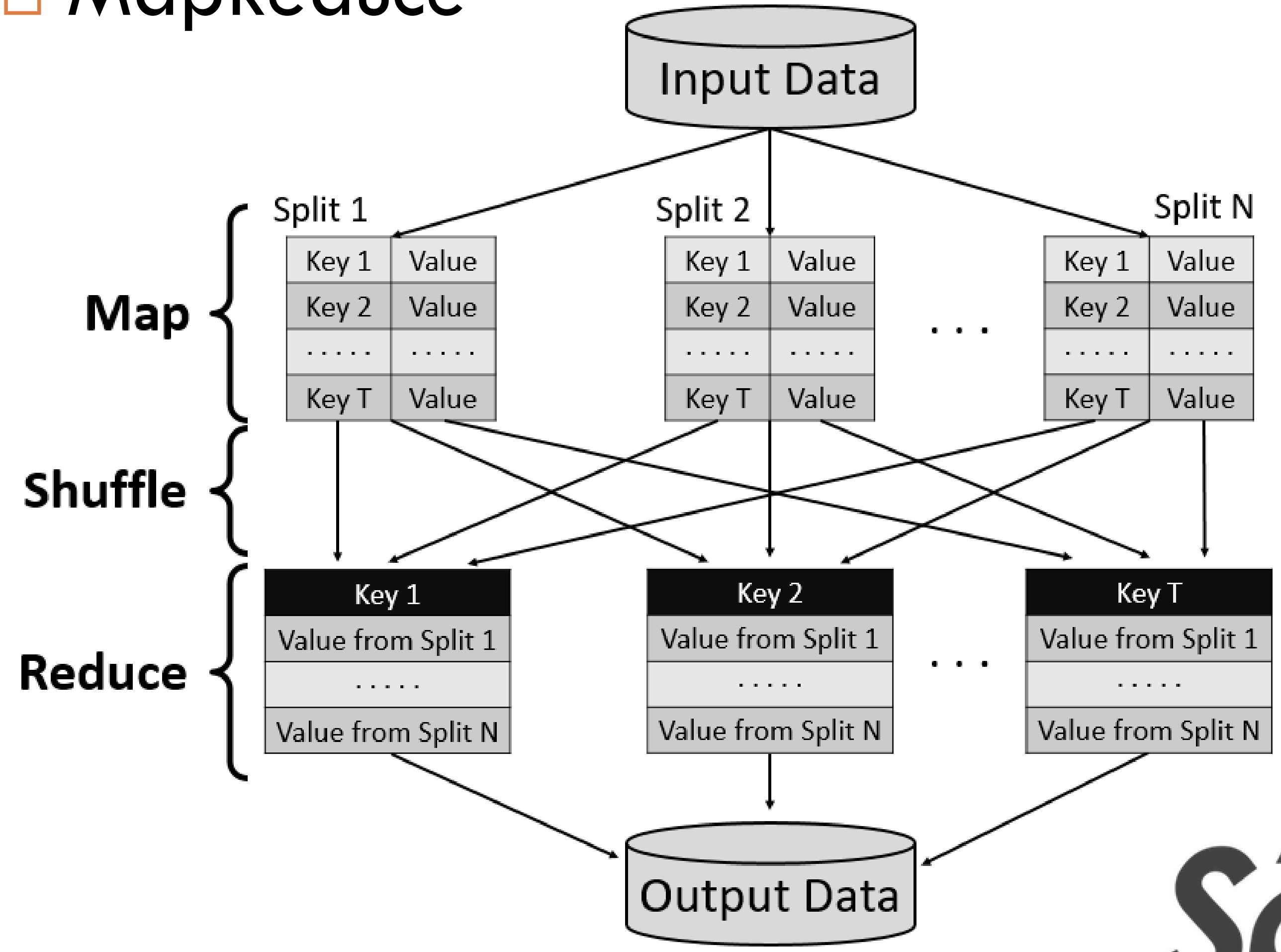
Big data

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



MapReduce

- □ MapReduce



Input Data

Map

Split 1		Split 2			Split N	
Key 1	Value	Key 1	Value		Key 1	Value
Key 2	Value	Key 2	Value	. . .	Key 2	Value
. . . . .	. . . . .	. . . . .	. . . . .		. . . . .	. . . . .
Key T	Value	Key T	Value		Key T	Value

Shuffle

Reduce

Key 1	Key 2		Key T
Value from Split 1	Value from Split 1		Value from Split 1
. . . . .	. . . . .	. . .	. . . . .
Value from Split N	Value from Split N		Value from Split N

Output Data

Outline

- □ Introduction & Preliminaries
- □ **kNN in the big data**
- □ kNN-IS: A MapReduce implementation for kNN under Apache Spark
- □ Fuzzy kNN for big data

kNN in the big data

The main problems to deal with large-scale data are:

- **Runtime:** The complexity of the traditional kNN is $O(n \cdot D)$, where n is the number of instances and D number of features.
- **Memory consumption:** For a rapid computation of the distances, the training set is normally stored in memory, what could easily exceed the RAM memory in the big data context.

Objective

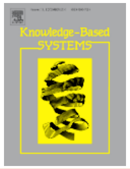
- □ The design of a **scalable kNN** approach that embraces the huge storage and processing capacity of cloud platforms, in order **to simultaneously classify** large amounts of unseen cases against a big (training) data set.
- □ To do so, we rely on the success of the **MapReduce framework.**

Outline

- □ Introduction & Preliminaries
- □ kNN in the big data
- □ **kNN-IS: A MapReduce implementation for kNN under Apache Spark**
- □ Fuzzy kNN for big data

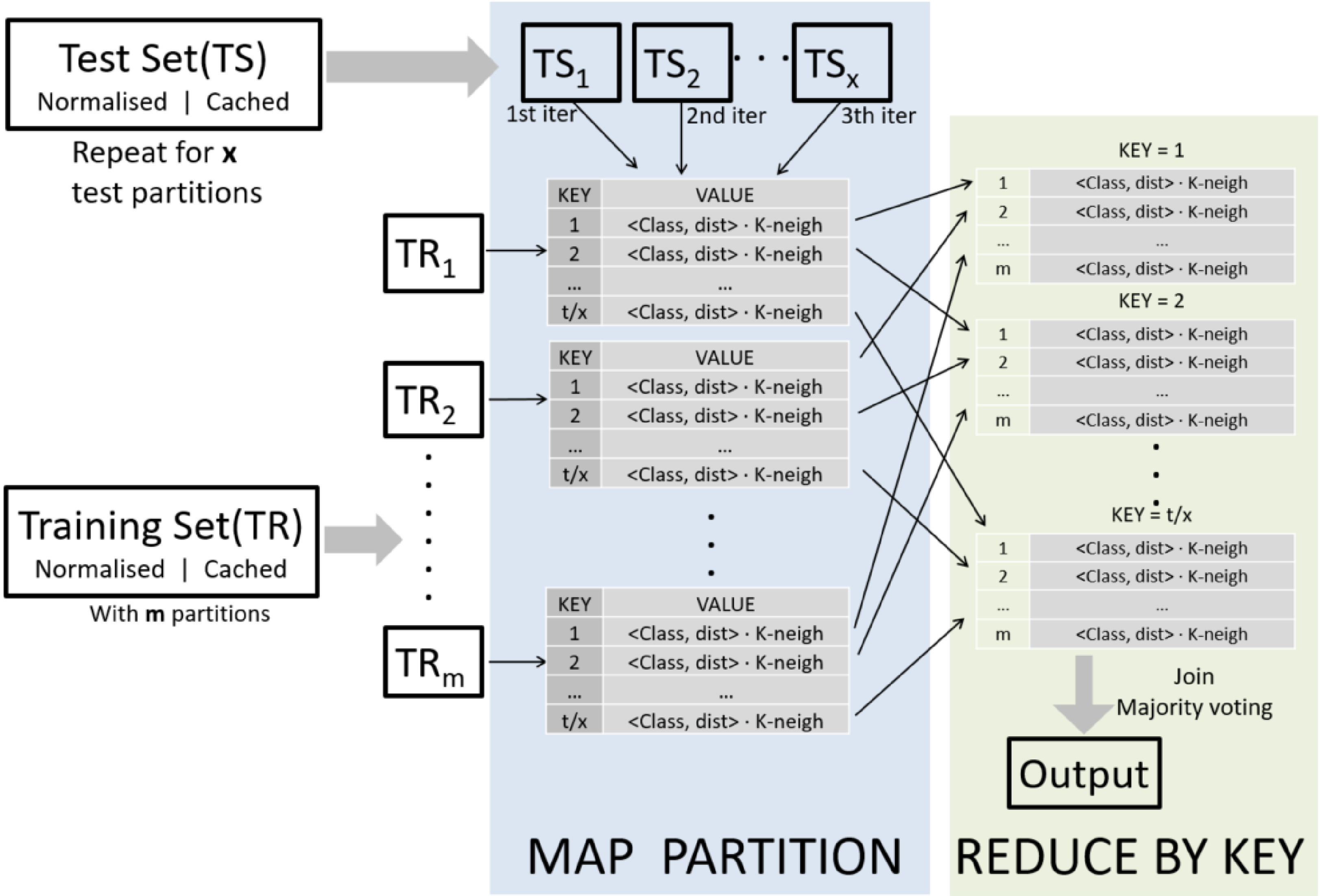
kNN-IS

- **Map phase:**
  - Split training set into m parts
  - Compute Class-Distance for each sample of test set against training samples
  - Split test set into n parts
- **Reduce phase:**
  - It groups all candidates to be the **k** closets neighbours
  - Calculate the real **k** nearest neighbours of each
  - It performs the majority voting and returns the predicted classes.



J. Maillo, S. Ramírez-Gallego, I. Triguero, F. Herrera. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowledge-Based Systems, 117 (2017) 3-15 doi: 10.1016/j.knosys.2016.06.012

kNN-IS: Flowchart



kNN-IS: Results

- The required runtime for the sequential version is very high (Susy dataset: 5 million instances)

Number of Neighbours (k)	Runtime (in minutes)	Accuracy (test)
1	54,314.15	0.6936
3	54,326.99	0.7239
5	54,419.76	0.7338
7	55,422.30	0.7379

- Around 908 hours or 37 days

kNN-IS: Scalability

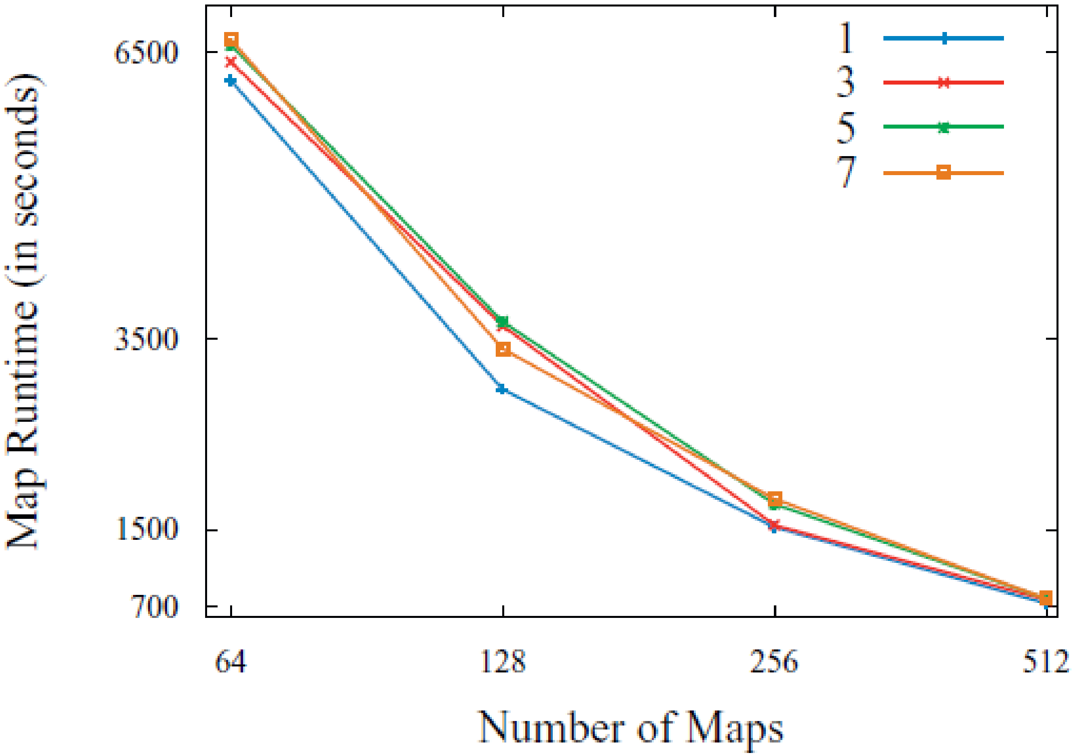
- Scalability with different number of neighbours

Number of Neighbours (k)	Runtime (in minutes)	Accuracy (test)
1	34.04	0.6936
3	38.30	0.7239
5	41.86	0.7338
7	41.96	0.7379

- From 37 days to 40 minutes

kNN-IS: Scalability

- Scalability with different number of maps



- Low runtime impact of the parameter of **k**

kNN-IS: Conclusions

- The same accuracy and very good achievements on runtimes
- The number of neighbours (*k*) does not drastically affect to the total runtime
- Deal with large training and test set when it exceeds the memory capacity by iterating
- The software can be found at SparkPackages

Outline

- □ Introduction & Preliminaries
- □ kNN in the big data
- □ kNN-IS: A MapReduce implementation for kNN under Apache Spark
- □ **Fuzzy kNN for big data**

Fuzzy kNN for big data

- The two main problems was increased:
  - **Runtime:** Two phases with the same complexity of the kNN-IS. Thus double the runtime.
  - **Memory consumption:** It is needed the training set twice plus the test set. More than kNN-IS

Fuzzy kNN: objective

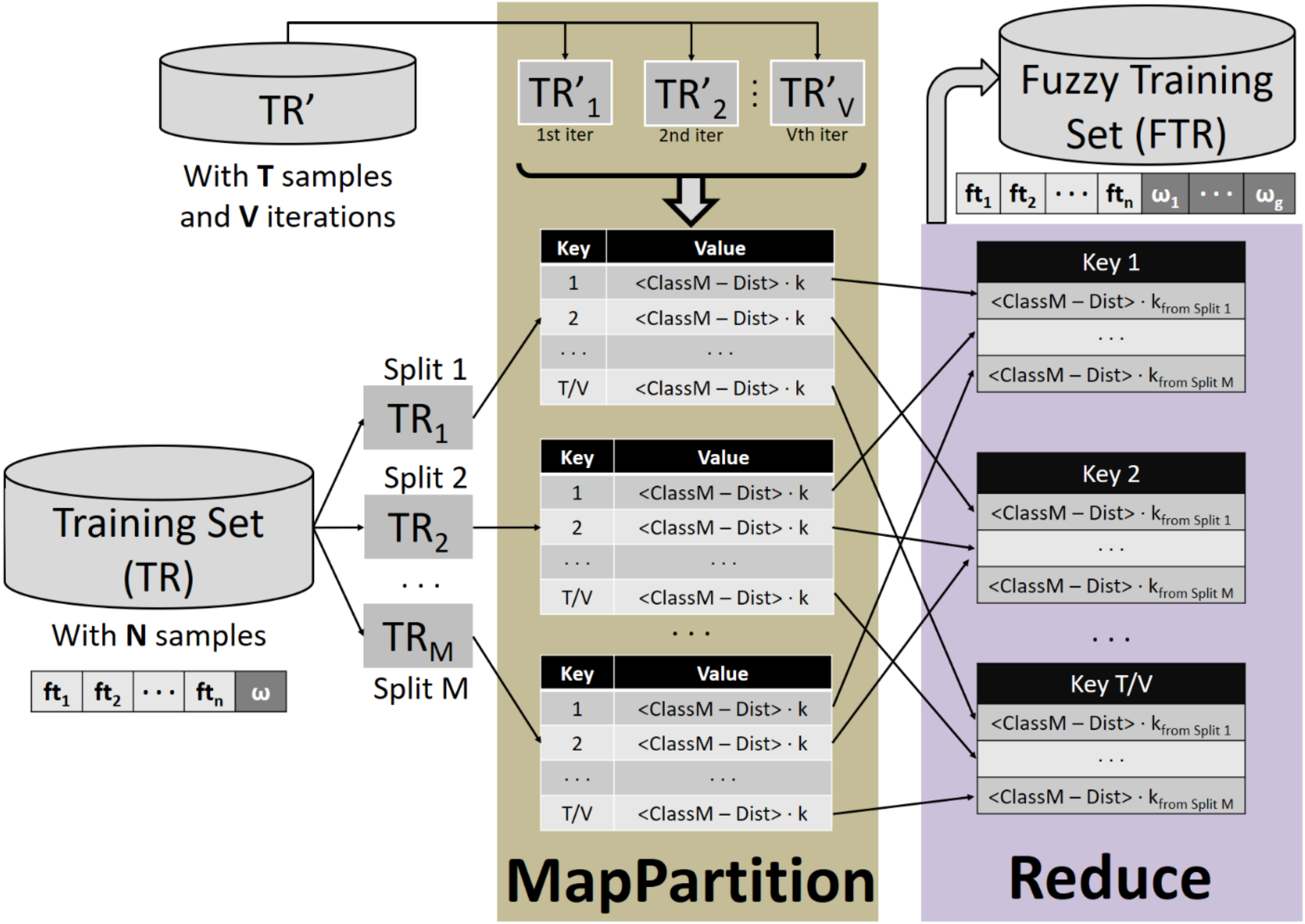
- The design of a **scalable Fuzzy kNN** approach that can manage large amounts of unseen cases against a big (training) data set

Fuzzy kNN: workflow

- **“Fuzzification” stage:**
  - Compute kNN-IS
  - Change class label → Class membership degree
  - Train vs Train
  - Output Fuzzy Training Set
- **Classification stage:**
  - Compute kNN-IS with the new Fuzzy Training Set
  - Select the class label with the higher membership degree

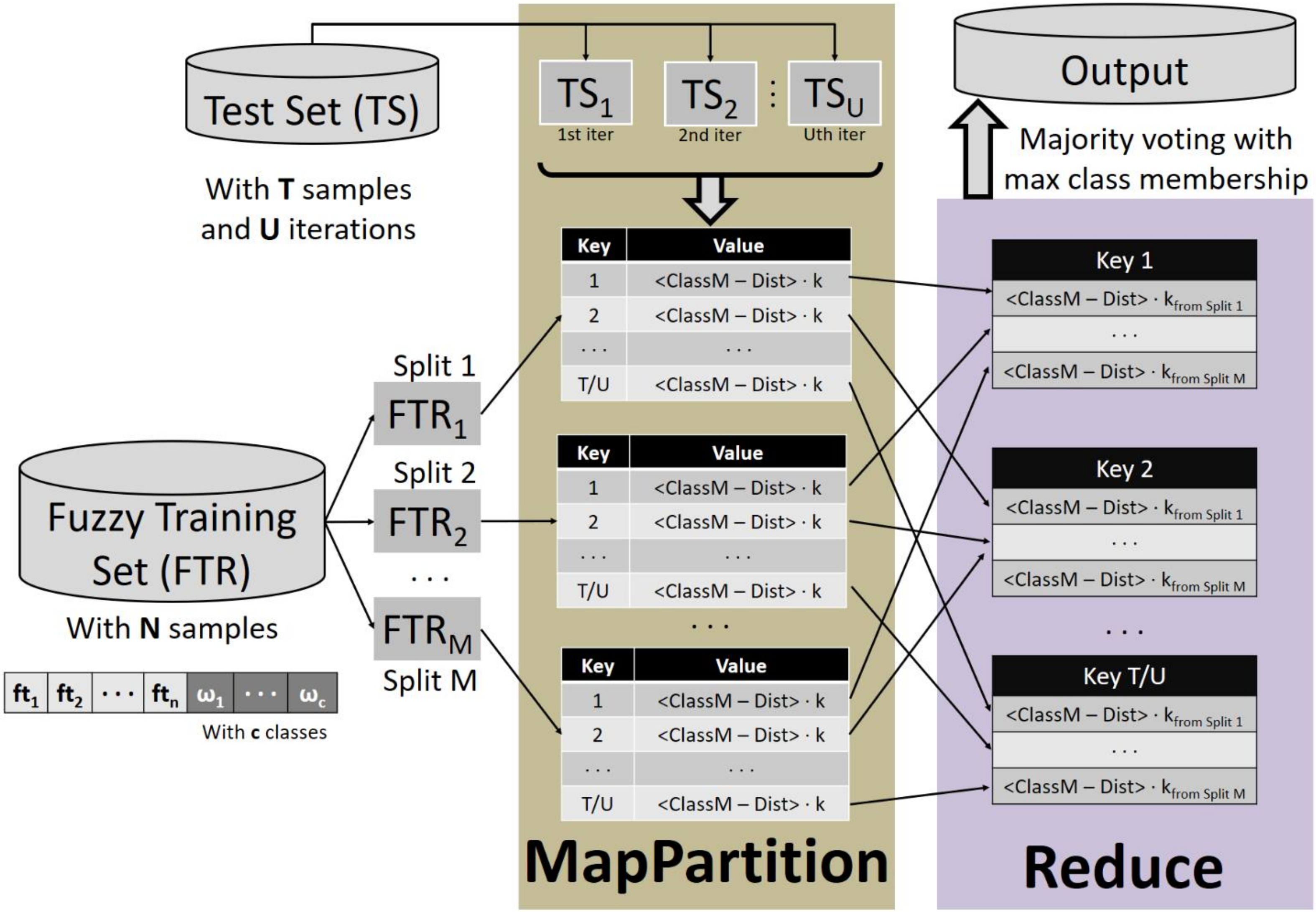
J. Maillo, J. Luengo, S. García, F. Herrera, I. Triguero. Exact Fuzzy k-Nearest Neighbor classification for big datasets. 2017 IEEE International Conference on Fuzzy System (FUZZ-iEEE) doi: 10.1109/FUZZ-IEEE.2017.8015686

Fuzzy kNN: “Fuzzification”



TR’
With **T** samples and **V** iterations
TR’$_1$
1st iter
TR’$_2$
2nd iter
:
TR’$_V$
Vth iter
Fuzzy Training Set (FTR)
ft$_1$ | ft$_2$ | · · · | ft$_n$ | ω$_1$ | · · · | ω$_g$
Training Set (TR)
With **N** samples
ft$_1$ | ft$_2$ | · · · | ft$_n$ | ω
Split 1
TR$_1$
Split 2
TR$_2$
. . .
TR$_M$
Split M
Key | Value
1 | <ClassM – Dist> · k
2 | <ClassM – Dist> · k
. . . | . . .
T/V | <ClassM – Dist> · k
. . .
Key 1
<ClassM – Dist> · k$_{\text{from Split 1}}$
. . .
<ClassM – Dist> · k$_{\text{from Split M}}$
Key 2
<ClassM – Dist> · k$_{\text{from Split 1}}$
. . .
<ClassM – Dist> · k$_{\text{from Split M}}$
. . .
Key T/V
<ClassM – Dist> · k$_{\text{from Split 1}}$
. . .
<ClassM – Dist> · k$_{\text{from Split M}}$
MapPartition
Reduce

Fuzzy kNN: Classify



Key	Value
1	<ClassM – Dist> · k
2	<ClassM – Dist> · k
· · ·	· · ·
T/U	<ClassM – Dist> · k

Key	Value
1	<ClassM – Dist> · k
2	<ClassM – Dist> · k
· · ·	· · ·
T/U	<ClassM – Dist> · k

Key	Value
1	<ClassM – Dist> · k
2	<ClassM – Dist> · k
· · ·	· · ·
T/U	<ClassM – Dist> · k

Key 1
<ClassM – Dist> · $k_{\text{from Split 1}}$
· · ·
<ClassM – Dist> · $k_{\text{from Split M}}$

Key 2
<ClassM – Dist> · $k_{\text{from Split 1}}$
· · ·
<ClassM – Dist> · $k_{\text{from Split M}}$

Key T/U
<ClassM – Dist> · $k_{\text{from Split 1}}$
· · ·
<ClassM – Dist> · $k_{\text{from Split M}}$

Fuzzy kNN: Results

Model	#Maps	Total Runtime (seconds)	Accuracy (test)
Exact Fuzzy kNN	256	285.34	0.7346
kNN-IS	256	38.30	0.7239

- Susy dataset with k = 3

kNN-IS: Conclusions

- Deal with large training and test set when it exceeds the memory capacity by iterating
- Depending on the situation, we will have time to get a better accuracy (**Exact Fuzzy kNN**) or faster runtimes (**kNN-IS**)
- Focus on the membership degree stage (bottleneck) with approximate kNN methods.

An exact scalable method of the k nearest neighbour algorithm to handle big data sets

BIG DATA II

Jesús Maillo (jesusmh@decsai.ugr.es)

08/03/2018