**Notes on Intuitionistic Fuzzy Sets**
**Print ISSN 1310–4926, Online ISSN 2367–8283**
**Vol. 24, 2018, No. 2, 129–135**
**DOI: 10.7546/nifs.2018.24.2.129-135**

# Big data, intuitionistic fuzzy sets and MapReduce operators

**Panagiotis Chountas[1], Krassimir Atanassov[2,3], Vassia Atanassova[2], Evdokia Sotirova[3], Sotir Sotirov[3] and Olympia Roeva[2]**

[1] University of Westminster Faculty of Science and Technology
Dept. of Computer Science
115 New Cavendish Street, London W1W 6UW, UK
e-mail: P.I.Chountas@westminster.ac.uk

[2] Dept. of Bioinformatics and Mathematical Modelling, Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., Sofia 1113, Bulgaria
e-mails: krat@bas.bg, vassia.atanassova@gmai.com, olympia@biomed.bas.bg

[3] Intelligent Systems Laboratory, Prof. Dr. Asen Zlatarov University
1 Prof. Yakimov Blvd, Burgas 8010, Bulgaria
e -mails: esotirova@btu.bg, ssotirov@btu.bg

**Received:** 20 March 2018 **Accepted:** 20 April 2018

**Abstract:** One of the main restrictions of the relational data model is the lack of support for flexible, imprecise and vague information in data encoding and retrieval. Fuzzy set theory and more specifically intuitionistic fuzzy sets provides an effective solution to model the data imprecision in relational databases. Several works in the last 30 years have used fuzzy set theory to extend relational data model to permit representation and retrieval of imprecise data. However, to the best of our knowledge, such approaches have not been designed to scale-up to very large datasets. In this paper, we develop *MapReduce* algorithms to enhance the standard relational operations with IFS predicates.
**Keywords:** Intuitionistic fuzzy sets, Big data, *MapReduce*.
**2010 Mathematics Subject Classification:** 03E72.

## 1 Introduction

The mainstream relational database queries use a Boolean logic to characterize users' answers. This means that the query condition is either satisfied or not satisfied. The use of Boolean logic

poses a restriction in expressing preference or ranking of query results. For instance, it seems quite natural for an online hotel-room search to answer questions such as: "*Give me all hotel-rooms which are not too expensive and are close to city centre*".

Intuitionistic fuzzy sets (IFS) [2, 3], and possibility theory [4] provide an effective solution to represent and process imprecise relational information. The IFS theory is an extension of the classical fuzzy set theory [12]. Each element of an intuitionistic fuzzy set has degrees of membership ($\mu$) and non-membership ($\nu$), which potentially do not sum up to 1.0 thus leaving a degree of indefiniteness or hesitation margin ($\pi$). As extension to the classical definition of a fuzzy set is given by

$$\tilde{A} = \{\langle x, \mu_A(x) \rangle \mid x \in X\}$$

where: $\mu_A(x) \in [0, 1]$ is the membership function of the fuzzy set $\tilde{A}$, an Intuitionistic fuzzy set $A$ is given by

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle \mid x \in X\}$$

where: $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1$ and $\mu_A(x), \nu_A(x) \in [0, 1]$ denote a degree of membership and a degree of non-membership of $x \in A$, respectively. Obviously, each fuzzy set may be represented by the following intuitionistic fuzzy set

$$A = \{\langle x, \mu_A(x), 1 - \mu_A(x) \rangle \mid x \in X\}$$

For each Intuitionistic fuzzy set in $X$, we will call $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ an intuitionistic fuzzy index (or a hesitation margin) of $x \in A$ which expresses a lack of knowledge of whether $x$ belongs to $A$ or not. For each $x \in A$, $0 \leq \pi_A(x) \leq 1$.

In [7, 8] we defined the main operations over intuitionistic fuzzy relations (IFR) such as projection, selection and join. Let $R$ be an (IFR), i.e.,

$$R = \{\langle x, \mu_R(x), \nu_R(x) \rangle \mid x \in X\}, \text{ where } x = \langle \text{col}_1, \ldots, \text{col}_n \rangle$$

is an ordered tuple belonging to a given universe $X$, $\{\text{col}_1, \ldots, \text{col}_n\}$ is the set of attributes of the elements of $X$, $\mu_R(x)$ is the degree of membership of x in the relation $R$. In other words, $R$ is an intuitionistic fuzzy subset of $X$ with membership and non-membership functions $\mu_R$ and $\nu_R$ respectively.

The *selection* operation defines a relation, which contains only those tuples from $R$ for which a certain predicate is satisfied. We can say that the selection modifies the degrees of membership and non-membership of $R$ depending on the corresponding value of the predicate:

$$\sigma_P(R) = \{\langle x, \min(\mu_R(x), \mu(P(x))), \max(\nu_R(x), \nu(P(x))) \rangle \mid x \in X\},$$

where $P$ is the predicate, i.e., the elements of the result relation have degree of membership, which is logically *AND*-ed with the corresponding value of the predicate $P$.

$$\textit{Project: } \prod_f \{\langle x, \mu_R(x), \nu_R(x) \rangle \mid x \in X\}.$$

The traditional *project* operator $\prod_f(R)$ selects all attributes $f$ from all tuples in $R$ leaving out other attributes not in $f$. The semantics of a intuitionistic fuzzy project operator $\pi_P\{\langle x, \mu_R(x), \nu_R(x) \rangle \mid x \in X\}$ should be that it selects all attributes $f$ from all *possibilities* in $R$. Projection does not affect the associated intuitionistic fuzzy measures.

$$\textit{Union: } P(A \cup B) = \{\langle x, \max(\mu_A(x), \mu_B(x)), \min(\nu_A(x), \nu_B(x)) \rangle \mid x \in X\}.$$

The intuitionistic fuzzy operator *union* merges two relations possibly containing possibilities for the same real world objects. To properly calculate the Intuitionistic fuzzy measures in the answer, it is beneficial to enumerate the possible worlds, i.e., consider each possibility of an element existing or not in the operand sets. The intersection and difference can be determined analogously.

A *Cartesian product* of two relations $R \times S$ is identical to the Cartesian product operation defined in the intuitionistic fuzzy sets theory [8], which uses the logical *AND* between the degrees of membership.

Let $S$ be another intuitionistic fuzzy relation: $S = \{\langle y, \mu_S(y), \nu_S(y) \rangle \mid y \in Y\}$, then:

$$R \times S = \{\langle\langle x, y\rangle, \min(\mu_R(x), \mu_S(y)), \max(\nu_R(x), \nu_S(y)) \rangle \mid \langle x, y\rangle \in X \times Y\}.$$

The definition of these operations is based on the notion of a probabilistic conjunction (logical *AND*). This type of conjunction is applied when the operands carry probabilistic semantics, i.e. they express a probability, not a degree of membership.

The focus of this paper is to develop *MapReduce* algorithms to scale-up the IFR operations to large scale crisp datasets. We formulate *t*-selection, projection, union, difference, intersection and Cartesian product operations with IFS predicates.

# 2 Operations on IFR queries with *MapReduce*

## 2.1 *MapReduce* framework

*MapReduce* is one of the most common platforms for processing big data, based their functioning on the primitive operations map and reduce [10], defined initially as part of the functional programming language, LISP. Many standard algorithms have been extended to comply [6, 9] with the shared-nothing architecture of *MapReduce* model of computation consists of two main functions: *Map* and *Reduce*. A large input file is broken into chunks and stored in a distributed file system. During the execution of a *MapReduce* job, the mapper tasks read an input split and call the *Map* function on each single record in the input split to produce a set of intermediate key-value pairs. The intermediate key-value pairs are hashed to one or more reducers based on their keys. The key-value pairs sent to each reducer are sorted and grouped by their keys. A reduce function is called for every key and all its associated values to produce a chunk of final output (Figure 1).

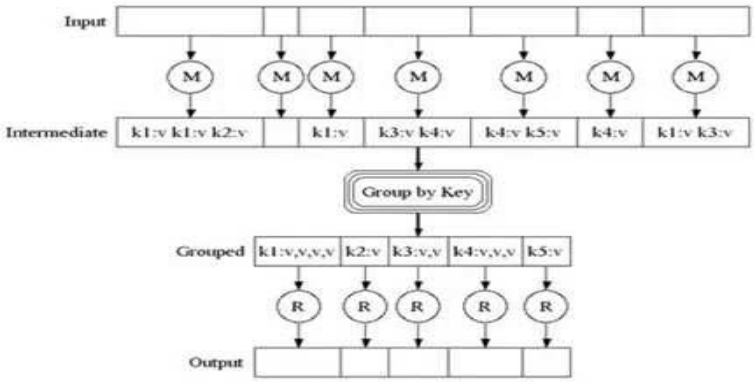


Figure 1. *MapReduce* Framework

The cost of a *MapReduce* job is expressed in terms of ($M$, $I$, $R$) where $M$ is the map cost across all records, $I$ is the communication cost of passing intermediate key-value pairs to the reducers and $R$ the total computation cost of all reducers.

## 2.2 Fuzzy relational operations in *MapReduce*

This section describes *MapReduce* algorithms for IFR operations on a large crisp dataset. The selection, union, intersection, and difference operations do not require much modifications to the crisp counterparts.

### *2.2.1. IFR selection*

The **selection** operator is a map-only job that reads a record $r$ from relation $R$, computes the degree $\langle \mu_R(x), \nu_R(x) \rangle$ to which $r$ satisfies a given condition and emits $r$ and $d = \min(\mu_R(x), \mu(P(x)))$, $\max(\nu_R(x), \nu(P(x)))$ as key and value, respectively.

**Algorithm-1 IFR selection map-only job**

**1 Map:**

Input: $r$ : an input record, $\langle \mu_R(x), \nu_R(x) \rangle$ membership & non-membership of $r$ in $R$,

$P$: Selection predicate

2: Parse $P$ and retrieve the fuzzy sets corresponding in $P$

3: compute $d = \min(\mu_R(x), \mu(P(x)))$, $\max(\nu_R(x), \nu(P(x)))$

4: emit key = $r$, value = $d$.

The cost of the selection operation is the cost of the *Map* function across all records, $O(|R|)$, where $|R|$ is the number of records in $R$.

### *2.2.2 IFR Projection*

The *Map* function for IFR projection $\pi_F(R)$ reads each input record $r$ in $R$ and emits the $F$ attributes of $r.F$, as key and the membership and non-membership degrese of $r$ in $R$ as values. A reducer receives a key $r.F$, produced by any of the map tasks, and a set of membership and non-membership degrees associated with it. It emits $r.F$and the maximum of its membership degrees and the minimum of its non-membership degrees.

**Algorithm-2 *MapReduce* job for IFR projection**

1: **Map:**

Input: $r$ an input record, $F$: projection attributes, $\langle \mu_R(x), \nu_R(x) \rangle$: membership of $r$ in $R$

2: emit key = $r.F$, value = $\langle \mu_R(x), \nu_R(x) \rangle$

1**: Reduce:**

Input: $u$: projected attributes of an input record, value = $\text{list}_{i:r_i.F=u} \langle \mu_R(r_i), \nu_R(r_i) \rangle$

2: emit key = $u$, values = $(\max_{i:r_i.F=u}(\mu_R(r_i), \min(\nu_R(r_i)))$

The cost of IFR projection is: $M + I + R = |R| + |\pi_F(R)| + |\pi_F(R)|$

### *2.2.3 IFR Union*

The *Map* function for fuzzy union $R \cup S$ reads an input record $r$ (from $R$ or $S$) and emits $r$ and its ⟨membership, non-membership⟩ degrees (in $R$ or $S$). The reduce function computes the maximum membership and minimum non-membership degree for each input record it receives as a key.

**Algorithm-3 *MapReduce* job for IFR $R \cup S$**

**1: Map:**

Input: $r$ : an input record, $\langle \mu(r), \nu(r) \rangle$ emit key = $r$, value = $\langle \mu(r), \nu(r) \rangle$

**1: Reduce:**

Input: $r$ : an input record, value = $list_i \langle \mu_i(r), \nu_i(r) \rangle$

2: emit key = $r$, value = ($list_i \langle \max(\mu_i(r)), \min(\nu_i(r)) \rangle$)

### *2.2.4 IFR Difference*

The *Map* function for IFR difference $R–S$ reads an input record $r$ and emits $r$ as the key. For value, it emits the name of the relation ($R$ or $S$) to which $r$ belongs as well as its membership and non-membership degree. The reduce function receives a record $r$ as the key and a list of its associated relations and degrees of membership in each relation. If $r$ belongs to $R$ but not $S$, it will emit $r$ and its membership degree in $R$. If $r$ belongs to both R and S it will emit $r$ and the minimum of its membership degrees in $R$ and $S$ complement.

Each reducer performs a linear operation on the values it receives. Hence, the total cost for fuzzy union, intersection and difference is:

$$M + I + R = O(|R| + |S|) + O(|R| + |S|) + O(|R| + |S|)$$

**Algorithm-4 *MapReduce* job for IFR difference $R–S$**

**1: Map:**

Input: $r$: an input record, $N$ name of the relation to which $r$ belongs ($R$ or $S$),

$\langle \mu_N(r), \nu_N(r) \rangle$ membership and non-membership of $r$ in $N$ respectiveley

2: emit key = $r$, value = ($\langle \mu_N(r), \nu_N(r) \rangle$, $N$)

**1: Reduce:**

Input: $r$ : an input record, value = list$N$ ($\langle \mu_N(r), \nu_N(r) \rangle$),

2: if value == $\{(\mu_R(r), \nu_R(r), R)\}$

then emit key = $r$, value = $\langle \mu R(r), \nu_R(r) \rangle$

end if

3: if value == $\{(\mu_R(r), \nu_R(r), R), (\mu_S(r), \nu_S(r), S)\}$ then

emit key = $r$, value = $\langle \min(\mu_R(r), \mu_S(r)), \max(\mu_R(r), \mu_S(r)) \rangle$

end if

### *2.2.5 IFR Join*

The fuzzy $R \otimes S$ join operation takes a IFR comparator (such as equal, greater than and fuzzy less than) and computes the degree to which every pair ($r \in R$, $s \in S$) satisfies the join condition.

**Algorithm-5 *MapReduce* job for IFR R⊗S**

**1: Map:**

Input: $r$ : an input record, $K$ name of the relation to which $r$ belongs ($R$ or $S$), $\langle \mu K(r), \nu K(r) \rangle$, $\alpha$: threshold value, $A$: the join attribute, A.min: a predefined lower bound on the values of $A$, $L$: length of $\alpha$-cut of IFS equal

2: if $\langle \mu K(r), \nu K(r) \rangle$ $\alpha$ then

3: $\text{Part}_1 = (r.A - A.\text{min}) / L$ then

5: emit key = $\text{Part}_1$, value = $\langle K,r,H_1 \rangle$

7: else

8: emit key = $\text{Part}_2$, value = $\langle K,r,H_2 \rangle$

9: end if

**1. Reduce:**

2.Value Input: = list $\langle (K,r,\text{flag}) \rangle$

3. emit key = $(v_1.r, v_2.r)$,

Value = $\langle \min(\mu v_1.K(v_1.r), \mu v_2.K\ (v_2.r)), \max(Vv_1.K(v_1.r), Vv_2.K(v_2.r)) \rangle$

The IFR join algorithm divides the domain of the join attribute into a number of ranges. Each range is called a partition and records from each relation get assigned to these partitions based on which range their join attribute value falls into. If the dataset is small, then one could also assume that each reducer could handle the data sent to it in short order. Unfortunately, all too often [1] neither is the case and some sub-partitioning of the data is needed to ensure load balancing.

# 3 Conclusions

This paper reports the implementation of a scalable flexible relational algebra in *MapReduce* based on IFS set theory. The IFS operations discussed are the IFR (selection, projection, union, difference and equal join. The cost of each algorithm is discussed in terms of the cost of the map and reduces functions as well as the communication cost. For future direction of this work, we will focus on two possible extensions of the current scalable IFR relational algebra specification.

The IFR join algorithm presented in this paper can only be applied when the join condition is fuzzy equal. An extension of the current framework will be considered when the join condition includes IFS greater than or less than comparators.

# Acknowledgements

This work has been supported by the Bulgarian National Science Fund under Grant Ref. No. DN-02-10 “New Instruments for Knowledge Discovery from Data, and their Modelling”.

# References


[1] Afrati, F. N., Sarma A. D., Menestrina, D., Parameswaran, A., & Ullman, J .D. (2012) Fuzzy joins using mapreduce. In: *IEEE 28th international Conference on Data Engineering (ICDE)*, 498–509.

[2] Atanassov K. (1986) Intuitionistic fuzzy sets, *Fuzzy sets and Systems*, 20(1), 87–96.

[3] Atanassov K. (1999) *Intuitionistic Fuzzy Sets*, Springer-Verlag, Heidelberg.

[4] Buckles, B. P., & Petry, F. E. (1982) A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7(3), 213–226.

[5] Chountas, P., Kolev B., Rogova, E., Tasseva, V., & Atanassov, K. (2007) *Generalized Nets in Artificial Intelligence, Vol. 4: Generalised Nets, Uncertain Data, and Knowledge Engineering*, Publishing House of Bulgarian Academy of Sciences, Sofia.

[6] Gufler, B., Augsten, N., Reiser, A. & Kemper, A. (2012) Load balancing in mapreduce based on scalable cardinality estimates. In: *IEEE 28th International Conference on Data Engineering (ICDE)*, 522–533.

[7] Kolev, B. (2004) Intuitionistic Fuzzy Relational Databases and Translation of the Intuitionistic Fuzzy SQL, In *Proceedings of the Sixth International FLINS Conference on Applied Computational Intelligence*, Blankenberge, Belgium, 189–194.

[8] Kolev, B. & Chountas, P. (2003) Intuitionistic Fuzzy Relational Databases, In:- *Proceedings of the Seventh International Conference on Intuitionistic Fuzzy Sets*, Sofia, Bulgaria, 109–113.

[9] Kyritsis, V., Lekeas, P., Souliou, D., & Afrati, F. (2012) A new framework for join product skew. In: Lacroix Z, Vidal M (eds) *Resource discovery*, vol 6799., *Lecture Notes in Computer Science*, Springer, NewYork, 1–10.

[10] McCarthy, J. (1960) Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I, *Communications of ACM*, 184–195.

[11] Rogova, E. & Chountas P. (2007) On Imprecision Intuitionistic Fuzzy Sets & OLAP - The Case for KNOLAP. *IFSA* (2), 11–20.

[12] Zadeh, L A. (1965) Fuzzy sets. *Information and Control*, 8(3), 338–353.