International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 9 5568 - 5573

Obtaining Approximation using Map Reduce by Comparing Inter-Tables

Prachi Patil
Computer Engineering Department, AISSMS, COE
Savitribai Phule Pune University
Pune, India
Prachimpatil114@gmail.com

Anilkumar Kadam
Computer Engineering Department, AISSMS, COE
Savitribai Phule Pune University
Pune, India
kadam_in@yahoo.com

***Abstract*—**Size of the data increasing day by day because of digital world at unpredictable rate. Basically size of raw data is increasing so deal with such rough data is the challenging task and we need to acquire knowledge from such a colossal data. Number of techniques are used to retrieve knowledge from raw data like genetic algorithm, fuzzy sets and rough set. Rough set is very popular method and basically depends upon approximation i.e. lower approximation, upper approximation, boundary region. Hence, the effective computation of approximation is important pace in improving the performance of rough set. There are number of ways to calculate this approximation. In our system we have calculated rough approximation independent of each other. This can be achieved by dividing input dataset, so that we can also reduce number of comparison. The division of dataset based on decision attribute in the dataset. In our paper, we have explained a new method for computing rough set approximation. Using map-reduce we can deal with massive data and able to compute rough approximation for massive dataset.

Keywords*-*Map; Reduce; Lower approximation; Boundary region; Upper approximation

*****

I. INTRODUCTION *(HEADING 1)*

From last few decennium, the size of the data stored in the databases has been increasing rapidly and therefore we face lots of difficulties for obtaining the worthful data. It has become very hard to reach exact and useful information as data stored in database growing each day. Data mining techniques are used to find out the rules and useful patterns within stored data. Storing huge amount of increasing data in the databases, i.e. information explosion, is necessary to transform these data into essential and useful information.

Data mining is a nontrivial process to determine valid, easily understandable dependencies in data. As the information technology field is developing, data volumes processed by many applications crossed the threshold in peta-scale [2], as a result it will in turn increase the computational requirements. Data processing and knowledge discovery [2] for colossal data is ever a burning topic in data mining. The big problem in data mining is the deficiency and indeterminateness. Such type of problems solved by using new procedures and theories, e.g. genetic algorithms, fuzzy sets, or rough sets etc.

Data mining used to extracting knowledge from large amount of data. It discovers interesting knowledge from large amounts of data stored either in data warehouse, database, or other information repositories.

II. RELATED WORK

Rough set theory is the most popular method in data mining. It is used to achieve rule generation from different datasets. Z.pawalak [13] [14] [15] invented rough set technique, describes about the importance of rule generation. In [1] author had achieved the parallel method for computing rough set approximation and tries to obtain result with proposing his algorithm to achieve lower and upper approximation. The effectual computation of rough set approximations is very important for improving the performance of data mining [2] [10].

Map Reduce terminology has been included in our research work so with referral to that many papers had describe about functionality of Hadoop file system [5]. So we have focused on map reduce terminology in below section of our paper. Initially some serial methods have been designed to achieve rough set approximation [6]. A massive data mining and knowledge discovery introduce a huge dispute with the growing data at an unpredicted rate [1]. Map Reduce is used to process big data at commodity hardware. It manages many large-scale computation [1] [12].

Algorithms corresponding to the parallel method based on the Map-Reduce technique are put forward to deal with the massive data. A. Pradeepa et al. [2], explained the purpose of data mining for big data, computing modes in parallel and algorithms are typical methods in research fields. Then the comprehensive result to evaluate the performances on the massive data sets show that demonstrated technology can effectively process of big data [3]. In paper [4], the mathematical principles of rough sets theory are explained and a sample application about rule discovery from a decision table by using different algorithms in rough sets theory is presented. In the document author described basic concepts of rough set and its advantage.

Rough set is a classifier [5] which has great importance in cognitive science and artificial intelligence, especially in machine learning, decision analysis, expert systems and inductive reasoning. It is also used to predict the malignancy degree of brain glioma [11].

III. ROUGH SET

Z. Pawlak has created a mathematical tool, rough set theory

in the beginning of the 1980s. It has been applied widely to extract knowledge from database [1]. It discovers hidden patterns in data. In decision making, rough set methods have a powerful essence in dealing with uncertainties. Rough sets can be used separately or combined with other methods such as statistic methods, fuzzy sets, genetic algorithms etc. The Rough set theory (RST) has been applied in several fields including data mining, pattern recognition, knowledge discovery, medical informatics, image processing etc.

In RST [7] [9], inconsistencies are not corrected or aggregated. In spite of this lower and upper approximations of all decision concepts are computed and using those rules get generated. Approximation perform vital role in performance of rough set theory. Because the rules are categorized into certain and approximate (possible) rules depending on the lower and upper approximations. Basically rough set is depend on approximation i.e. upper approximation and lower approximation and boundary region as mentioned below which is calculated later in this paper. Approximations are fundamental concepts of rough set theory.

- **Lower approximation** – The lower approximation consists of all the objects without any ambiguity based on attributes. These are surely belong to the set. In another way, we can say that, objects in lower approximation have only one decision for corresponding condition attribute value.

- **Upper approximation** – The objects are probably belong to the set, cannot be described as not belonging to the set based on the knowledge of the attributes. It contains all objects which possibly belong to the set. In another way, we can say that, objects in upper approximation are union of lower approximation and boundary region of corresponding decision value.

- **Boundary region** – Boundary region consist of the all objects having same condition attribute value but different decision value. In this set, we can get one than one decision value for same condition attribute value.

Pawlak suggested two numerical measures of imprecision of rough set approximations as,

_ Accuracy measure

_ Roughness measure

Accuracy measure is the ratio of the lower approximation of decision to the upper approximation of corresponding decision. Mathematically is can be defined as,

Where,

$$\alpha_D = \text{lower}_D(X) / \text{upper}_D(X).$$

X is dataset,

$X \neq \emptyset$; and

|.| denotes the cardinality of set.

$0 <= \alpha_D <= 1$

Based on the accuracy measure, the roughness measure is defined as,

$$\alpha D(X) = 1 - \alpha D(X)$$

If boundary region of any set is empty, then the set is called crisp set [16]. It means that all the objects in set has unique decision value for every condition. If the boundary region of set is nonempty, then it is called as rough set. In this set, for same condition value we can have more than one decision value. Rough set deals with vagueness and uncertainty which is most important in decision making. Data mining is a field that has an important contribution to data analysis, discovery of new meaningful knowledge, and independent decision making [15]. The rough set theory offers a feasible approach for decision rule extraction from data. Rough set theory (RST) employed mathematical modeling to deal with class data classification problems, and then proved to be a very useful tool for decision support systems, particularly when hybrid data, vague concepts and uncertain data were involved in the decision process [6].

Let T is decision table and *T = (U, A)* where U is universal set and A be the set of attributes. If $B \subseteq A$ and $X \subseteq U$ We can approximate *X* using only the information contained in *B* by constructing the *B-lower*, *B-upper* approximations and boundary region of *X*, denoted $\underline{B}X$, $\overline{B}X$ and BR respectively, where,

$$\underline{B} = \{x | [x]_B \subseteq X\}$$
$$\overline{B} = \{x | [x]_B \cap X \neq \emptyset\}$$
$$BR = \overline{B} - \underline{B}$$

There are many application related with massive data as text mining, association rule mining, temporal data mining, and sequential pattern mining and many more algorithms based on RST.

IV. System Architecture

The system architecture is as shown in the following block diagram. In the system user has choice to select desired condition attributes. The selected input dataset need to move from local machine to HDFS. So there program execution takes place.

A. System Architecture

In Proposed system we can compute rough set approximation without computing equivalence classes, decision classes and associations between them using the Map-Reduce technique as existing system. Lower and upper approximations are computed by comparing sub tables generated by input table. The system architecture is as shown in figure 1.

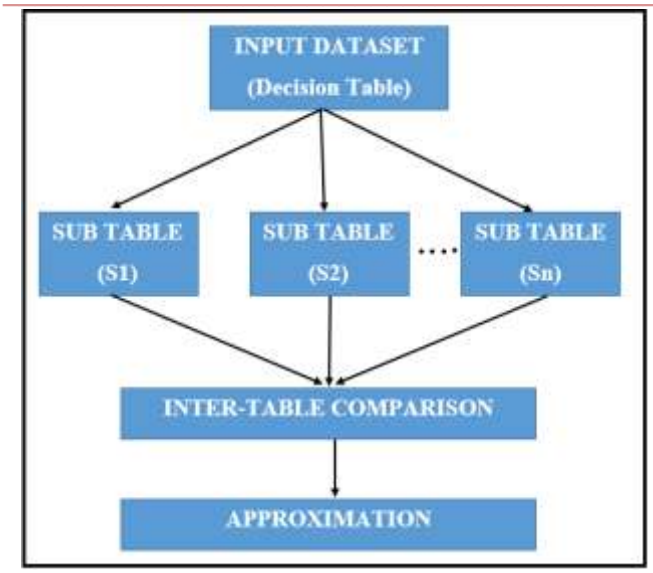

Figure 1. System Architecture

B. Mathematical Model

Input: Dataset

Output: Lower Approximation, Boundary Region

Proposed System:

S = (U, A, V, f).

Where,

S = decision table

U = f x1, x2,. . . , xn}

: Non-empty finite set of objects (universe)

A = {a1, a2, . . . , am}

: Non-empty finite set of attributes (features)

A = C U D,

C: set of condition attributes

D: set of output or decision results

C ∩ D = ∅

V: value of attributes

f: U × A → V is an information function such that function f(x, a) ∈ Va

Divide decision based on decision attribute value:

No. of sub tables = domain of decision results

Divide S into {S1, S2, S3, _ _ _ _ _ _ Sn}

Where,

n= Domain (D)

Compare (f(x,C), f(y,C))

Where,

x ∈ Si, y ∈ Sj and

Si ∈ S and Sj ∈ S.

i ≠ j, 1 <= i <= n and 1<=j <= n.

C. Block Diagram of System

The most important step in the system is to divide dataset on the basis of decisions. Hence number of sub tables is equal to the domain of decision attribute from the dataset. Here, we are reducing the computation as well as number of comparison. So this is the key step in the system. One more advantage is we can get boundary region independent on lower approximation and upper approximation. In the existing we need to compute lower approximation and upper approximation to get boundary region. Hence we get approximation simply by performing inter table comparison.

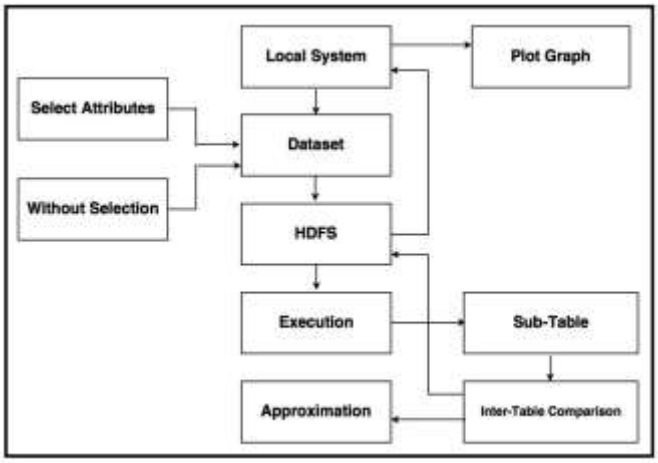

Figure 2. Block Diagram

D. Example for System

In this section we have taken sample dataset to compute approximation based on inter table comparison. As we are going to deal with rough set, we need to consider dataset as decision tables. Decision table mathematically represented as, S= (U, A, V, f), where U = {x1, x2,. . . , xn} is a universe, which is non-empty finite set of objects. A = {a1, a2, . . . , am} is a non-empty finite set of attributes (features). S = (U,A,V, f) is also called a decision table if A = C ∪ D, where C is a set of condition attributes and D is a decision attribute, C ∩ D = ∅. V = $\bigcup_{a \in A} V_a$. Va is a domain of the attribute a. An information function is f, where, f: U ×A→V. f(x,a) ∈ Va for every x ∈ U, a ∈ A. f(xi,aj) denotes the value of object xi on attribute ai. Consider the example as shown in table 1.

TABLE I. DECISION TABLE

Object	a1	a2	D
x1	0	0	0
x2	0	0	1
x3	1	0	1
x4	1	1	1
x5	0	0	0
x6	1	1	2
x7	0	0	0
x8	1	1	1
x9	1	1	2
x10	0	1	1
x11	1	0	2
x12	1	1	1

For data set in table 1,
S= {U, A, V, f}
Where,
U={ x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12}
A= CUD
C= {a1, a2}
D= {D}
V= {$V_{a1,}$ V_{a2}, V_D}
V_{a1}= {0, 1}
V_{a2}= {0, 1}
V_D= {0, 1, 2}

Unlike the existing system, we will divide the input dataset based on the decision. Hence number of sub data sets equal to the number of domain of decision attribute. Divide S = {U, A, V, f} into Si. Where 1<=i<=Domain of decision attribute. As a result we will get Si = {Ui, A, V, f}. Such that, S=$\bigcup_{i=1}^{d(D)} Si$ and U=$\bigcup_{i=1}^{d(D)} Ui$. In dataset of table 1, domain of D, d (D) =3. i.e. {0, 1, 2}. Therefore we will get S1, S2, S3 as shown in table 2
.

TABLE II. DIVIDE DECISION TABLE BASED ON DECISION VALUE

DECISION TABLE FOR D=0

Object	a1	a2	D
x1	0	0	0
x5	0	0	0
x7	0	0	0

DECISION TABLE FOR D=1

Object	a1	a2	D
x2	0	0	1
x3	1	0	1
x4	1	1	1
x8	1	1	1
x10	0	1	1
x12	1	1	1

DECISION TABLE FOR D=2

Object	a1	a2	D
x6	1	1	2
x9	1	1	2
x11	1	0	2

As shown in table 2,
S1= {U_1, A(C, D_1), V, f}
S2= {U_2, A(C, D_2), V, f}
S3= {U_3, A(C, D_3), V, f}
Where,
U_1= {x1, x5, x7}
D_1= {0}
U_2 = {x2, x3, x4, x8, x10, x12}
D_2= {1}
U_3= {x6, x9, x11}
D_3= {2}

We will maintain two sets one for lower approximation and another for boundary region. After that compare the values of condition attributes of each object from one sub table to the each object of remaining sub tables under two conditions. First if values of object in one sub table matches with the value of one or more objects from one or more sub tables, we will put all objects in the set of boundary region. Second, if values of object in one sub table does not matches with value of any object from remaining table, we will put that object in the set of lower approximation. We get the output as described in section F. Result.

All this steps we can present in the form of activities. We need to follow steps of intra table comparison as shown in figure 3 to get the output as approximation.

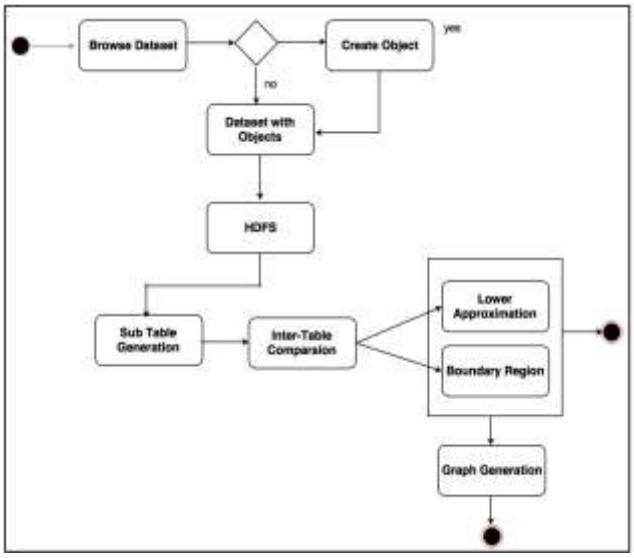

Figure 3: System Activity

As shown in figure 3, system Activity diagram describes dynamic aspects of the system. It is a basic flow chart to represent the flow form one activity to another activity. Such that initially data is browsed then whether to create the objects or the dataset already has it that decision took place. After that it has been copied to Hadoop file system. Then we create the sub table from the given dataset and after that we compare it on the basics of comparison we decide whether the instance comes under lower approximation or under upper approximation. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.

E. Algorithm

Steps to Compute By Proposed Method

Input:

 ISSN: 2321-8169
5568 - 5573

Decision table (data set)

Output:

Rough Approximation

Method:

Step 1:

Choose the data set

Step 2:

Divide the dataset (decision table) based on decision. No of sub tables = domain of decision attribute in dataset

Step 3:

Compute approximation for every decision (compare inter table objects)

i) If value of object in one table matches with value of objects in another table, add both objects in boundary region of corresponding decision.
ii) If value of object in one table does not matches with any value of objects in another tables, add that object in lower approximation of corresponding decision.
iii) Upper approximation is union of lower approximation and boundary region.

Step 4:

Compute approximation for entire dataset.

F. Block Diagram of System

To compute approximation of each decision value, if we found the same values of condition attributes of objects into another sub table. We will add all object into the boundary region of corresponding decision. Example: for the first step, condition attribute values of x1 is (0, 0), which is same as condition values of attribute of object x2 in another table. Hence we will add both of them in boundary region of corresponding attribute as Boundary region (D=0) ={x1, x2} and boundary region of (D=1) ={x1, x2}. Likewise we will add the objects into these sets as condition satisfies.

Finally we will get,

Boundary region (BR) of for every decision as,
BR (D=0) = {x1, x2, x5, x7},
BR (D=1) = {x1, x2, x3, x4,x5,x6,x7,x8,x9,x11,x12},
BR (D=2) = {x3, x4, x5, x8, x11, x12}.

Lower approximation as,
L (D=0) = {Ø},
L (D=1) ={x10},
L (D=2) = {Ø}.

Decision / Appro-Ximation	D=0	D=1	D=2	Dataset
Lower Approx	{}	{x10}	{}	{x10}
Boundary Region	{x1,x2, x5,x7}	{x1,x2,x3,x4,x5,x6, x7,x8,x9,x11,x12}	{x3,x4, x5, x8, x9,x11 ,x12}	{x1,x2,x3, x4,x5,x6, x7,x8,x9, x11,x12}
Upper Approx	{x1,x2 ,x5,x7}	{x1,x2 ,x3,x4, x5x6 ,x7,x8, x9,x10 x11,x12	{x3,x4, x5 ,x8, x9,x11, x12}	{x1,x2, x3,x4, x5x6, x7, x8, x9,x10, x11 ,x12}

Upper approximation as,
U (D=0) = {x1, x2, x5, x7},
U (D=1) = { x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12}.
U (D=2) = {x3, x4, x5, x8, x11, x12}.

This output is shown in tabular format in result section.

V. MAP REDUCE

Map-Reduce [8] allows for distributed processing of the Map and Reduce functions [2]. The Map-Reduce divides the input file into no of blocks by "input split" method. Map reduce is used for processing data on commodity hardware.

Our system follows this steps of HDFS to achieve the respective output. Initially input has been as text file on which operation has been performed to achieve the results.

The rough set approximations obtained by the parallel method are the same as those obtained by the serial method. But using map reduce we can run independent phases in parallel based on map-reduce. Therefore time required is very less as compared to traditional method of rough set calculation. In addition to that we can also generate the rules for massive data and able to abstract attributes in more efficient way using map-reduce with rough set.

Data partitioning, fault tolerance, execution scheduling are provided by MapReduce framework itself. MapReduce was designed to handle massive data volumes and huge clusters. MapReduce is a java programming framework that allows to execute user code in large cluster. All the user has to write two functions: Map and Reduce. During the Map phase, the input data are distributed across the mapper, where each machine then processes a subset of the data in parallel and produces one or more <key; value> pairs for each data record. Next, during the Shuffle phase, those <key, value> pairs are repartitioned (and sorted within each partition) so that values corresponding to the same key are grouped together into values {v1; v2; ……}. Finally, during the Reduce phase, each reducer machine processes a subset of the <key, {v1; v2; ….}> pairs in parallel and writes the final results to the distributed file system. The

map and reduce tasks are defined by the user while the shuffle is accomplished by the system. the map and reduce functions supplied by the user have associated types.

Map (k1, v1) → list (k2, v2)

Reduce (k2, list (v2)) → list (v2)

Two programmer specified functions:

- Map
  Input: key/value pairs i.e. (k1, v1)
  Output: intermediate key/value pairs i.e. list (k2, v2)
- Reduce
  Input: intermediate key/value pairs i.e. (k2, list (v2))
  Output: List of values list (v2)

That is, the input keys and values are drawn from a different domain than the output keys and values. The k1, k2 are the two different keys used in MapReduce phase and same as v1, v2 are the different values. The intermediate keys and values are from the same domain as the output keys and values. These keys and values are obtain from the dataset which we browse to process our algorithm and get respective output.

Map-Reduce framework offers clean abstraction between data analysis task and the underlying systems challenges involved in ensuring reliable large-scale computation [16]. Map-Reduce runtime system can be transparently explore the parallelism and schedule components to distribute resource for execution.

VI. CONCLUSION

Many rough sets based algorithms have been developed for data mining. But enlarged data in applications made these algorithms based on RST a challenging task. Computation of rough set approximation is very important step. We can improve the quality and speed of calculating approximation. This is one way where we have lots of opportunities to achieve speed and accuracy. Proposed parallel method for rough set computes rough approximation in more efficient way as compare to existing one.

REFERENCES

[1] Junbo Zhang a, Tianrui Li , Da Ruan, Zizhe Gao, Chengbing Zhaoa, ' A parallel method for computing rough set approximations', J. Zhang et al. / Information Sciences 194 (2012) 209–223

[2] A.Pradeepa 1, Dr. Antony Selvadoss Thanamani Lee2 ,"Hadoop file system and Fundamental concept of Mapreduce interior and closure Rough set approximations", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013

[3] 'Rough set based decision support' Roman Slowinski Institute of Computing Science Poznan University of Technology and Institute for Systems Research Polish Academy of Sciences Warsaw, Poland

[4] Mert BaMathematical Engineering Department, Yildiz Technical University, Esenler, İstanbul, TURKEY 'Rough Sets Theory as Symbolic Data Mining Method: An Application on Complete Decision Table'

[5] Mehran Riki , Hassan Rezaei University of Sistan and Baluchestan, Zahedan, IRAN Department of Mathematics, 'Introduction of Rough Sets Theory and Application in Data Analysis'.

[6] Agnieszka Nowak – Brzezinska ' Rough Set Theory in Decision Support Systems'

[7] Xu, Jager, H.P. Kriegel, "A fast parallel clustering algorithm for large spatial databases, Data Mining and Knowledge Discovery" (1999) 263–290.10.1023/ A: 100988480934).

[8] Y.K. Patil, Prof. V.S. Nandedkar "Hadoop: New Approach for Document Clustering", International Journal of Advanced Research in IT and Engineering. ISSN: 2278-6244.

[9] Ing. Pavel JURKA, Dr. Frantiek Zboril "Using Rough Sets In Data Mining", IEEE transactions on knowledge and data engineering, vol. 24, no. 10, October 2012.

[10] Silvia Rissino, Germano Lambert-Torres "Rough Set Theory Fundamental Concepts, Principals, Data Extraction, and Applications, Source: Data Mining and Knowledge Discovery in Real Life Applications", ISBN 978-3-902613-53-0, pp. 438, February 2009, I-Tech, Vienna, Austria.

[11] Yuhua Qian, Jiye Liang, Witold Pedryczb, Chuangyin Dang "Positive approximation: An accelerator for attribute reduction in rough set theory", Artificial Intelligence 174 (2010) 597618 Elsevier 2010.

[12] Ping ZHOU, Jingsheng LEI, Wenjun YE, "Large-Scale Data Sets Clustering Based on MapReduce and Hadoop", Journal of Computational Information Systems 7: 16 (2011) 5956-5963.

[13] Zdzisaw Pawlak, Andrzej Skowron "Accelerator for attribute reduction in rough set theory Rudiments of rough sets", Information Sciences 177 (2007) 327.

[14] Zdzisaw Pawlak, Andrzej Skowron,"Rough sets: Some extensions", Information Sciences 177 (2007) 2840.

[15] Zdzisaw Pawlak, Andrzej Skowron, "Rough sets and Boolean Reasoning", Information Sciences 177 (2007) 4173.

[16] Prachi Patil, "Data Mining with Rough Set Using MapReduce" International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) ISSN(Online) : 2320-9801 Vol. 2, Issue 11, November 2014.