Approximate Query Processing on High Dimensionality Database Tables Using Multidimensional Cluster Sampling View

Tomohiro Inoue*, Aneesh Krishna*, Raj P. Gopalan

Department of Computing, Curtin University, Bentley 6102, Western Australia, Australia.

* Corresponding author. Email: tomohiro.inoue@postgrad.curtin.edu.au, a.krishna@curtin.edu.au

Manuscript submitted October 16, 2015; accepted November 30, 2015.
doi: 10.17706/jsw.11.1.80-93

Abstract: Approximate query processing based on random sampling is one of the most useful methods for the efficient computation of large quantities of data kept in databases. However, small samples obtained through random sampling methods might lack the appropriate data relevant to query conditions because the samples do not adequately represent the entire dataset. The Multidimensional Cluster Sampling View has been proposed to support efficient and effective approximate query processing on common database tables. This view provides random sample records to be drawn from a database in SQL efficiently and effectively. The effectiveness of approximate query processing in this view was demonstrated on a large database table with only four dimensions. This differed from the usual number of dimensions in decision support systems, which is most commonly over ten. Therefore, further examinations and evaluations focusing on dimensionality, such as ten-dimensional data and over, are required in order to demonstrate its practicality. This paper evaluates whether the number of dimensions have an impact on the accuracy of the approximation and on the performance of the Multidimensional Cluster Sampling View. The results of the evaluation show that the effects of dimensionality are not visible.

Key words: Approximate query processing, databases, data warehouses, decision support systems, dimensionality, indexing, sampling.

1. Introduction

With an ever-increasing accumulation of enterprise data, traditional query processing, which provides precise answers, is time-consuming when used in decision support systems with large quantities of data [1]-[7]. However, it is unnecessary for decision makers to obtain absolute answers; approximate answers which are quite close to the exact answers can be acceptable in the decision-making process [1], [3], [8]. Approximate query processing based on random sampling is a very efficient method for dealing with extensive volumes of data [4], [8]-[14]. The sampling-based approximate query processing can achieve fast data processing in large databases due to only requiring access to a small amount of data. However, a small sample in highly selective queries or skewed data distribution is likely to miss most, if not all, of the relevant data because the small sample may not be fully represented throughout the entire dataset [5], [8], [11], [13]. A large sample can attain a higher level of confidence; however, its computation is much more time-consuming. As a result, the large sample negates the benefits of sampling.

To solve the trade-off problem, the ACE tree has been proposed by Joshi and Jermaine [12]. It is a binary tree index structure designed to draw random sample records relevant to the query condition. However, the height of the index tree becomes excessive in the high dimensional data. Rudra, et al. [8] have proposed the k-MDI tree as a solution to the problem. The k-MDI tree is a k-ary balanced index tree, extending the ACE tree. Seeing as the height of the k-MDI tree is limited to the number of dimensions, the k-MDI can achieve a more efficient index search. However, the feasibility of the k-MDI in terms of actual implementation was not shown, and the search algorithm present in the k-MDI cannot provide sufficient samples under a certain query condition.

To solve the problems of the k-MDI, the Multidimensional Cluster Sample View, based on the k-MDI, was proposed in [15], which is a hybrid view with an index for efficient and effective approximate query processing. The view can be simply implemented into two database tables. Utility software was developed, enabling the Multidimensional Cluster Sampling View to be built on database tables with a few parameters and to execute approximate query processing with ease in this view. The approximate query processing in this view provides a high level of accuracy when used on a large dataset. It is empirically demonstrated that approximate answers with high quality (less than 1.5% error) can be provided on a large table with four dimensions using only 1% samples, even in a highly selective query which occupies only 0.1% of the complete set of data. Also, the speed of query processing in this view is significantly faster than full scans and a columnar database called InfiniDB. Furthermore, significant improvement of query performance between 10 and 99 times that of the full scans is consistently demonstrated through empirical evidence. In comparison with the columnar database, the improvement of query performance reaches up to 30 times in the experiment.

In general, multidimensional models in data warehouses comprise of approximately 8 to 15 tables [16]. This means that approximate query processing used in decision support systems must work properly over ten-dimensional key attributes for practical use. Therefore, the Multidimensional Cluster Sampling View is needed to perform further evaluations in terms of dimensionality including high-dimensional data. To evaluate the Multidimensional Cluster Sampling View with multi-dimensional data, there are three points which must be examined, namely accuracy of approximation, query performance and construction performance. However, the second point, the query performance of approximate query processing with various dimensions, has been already evaluated with a columnar database called InfiniDB in [15]. It has been demonstrated that the number of dimensions does not affect the query performance in the view. Hence, the rest of the points are critical to demonstrate practicality of the Multidimensional Cluster Sampling View in data warehouses.

In this paper, the effects of dimensionality on accuracy in the Multidimensional Cluster Sampling View are examined and evaluated. Furthermore, an analysis of the effects of dimensionality on the construction performance of the Multidimensional Cluster Sampling View is also carried out.

The rest of this paper is organised as follows: in Section 2, the terms and definitions used in this paper are provided. Section 3 examines and evaluates the effects of the number of dimensions on accuracy in the Multidimensional Cluster Sampling View. In Section 4, the impact of the number of dimensions on the construction time of this view is examined and evaluated. Finally, Section 5 concludes the paper and indicates some directions for further research.

2. Terms and Definitions

2.1. Multidimensional Cluster Sampling View

The Multidimensional Cluster Sampling View, which is proposed in Inoue *et al.* [15], is a view with an index for approximate query processing on large databases. The concept is based on combining set theory and the k-MDI structure proposed by Rudra *et al.* [8]. As shown as an example in Fig. 1, the conceptual image is made up of a number of random data sets called Cluster Samples. Each Cluster Sample contains a leaf number (L_n), a section number (S_n), data range property ($KeyAttribute.Range$), size property (N), and a random sample. In this example, there are three key attributes (dimensions), which are the DATE attribute divided into three range partitions, ITEM attribute divided into three range partitions, and STORE attribute divided into three range partitions. The leaves L_n are created by all combinations of the partitions. In other words, the number of leaves L_n is the direct product of the number of partitions on each key attribute, namely 27 $(3\, partitions\ on\ DATE \times 3\, partitions\ on\ ITEM \times 3\, partitions\ on\ STORE)$ in this example. Each leaf is composed of several sections containing a random sample, which are equal in number to the key attributes plus one, leaving namely $(3 + 1 = 4)$ sections in this example. The combinations of all data ranges are assigned to the last section number, $L.S_4$, in each leaf. Therefore, all tuples in section 4 $(L.S_4)$ are within specified data ranges for all key attributes. All tuples in Section 3 $(L.S_3)$ are within specified data ranges for only the first and second key attributes. The sample records in section 2 $(L.S_2)$ are also formed in the same manner, with only the tuples for the first key attribute being contained within a specified range. Section 1 $(L.S_1)$ contains a random sample for all key attributes, without any restriction of data range.

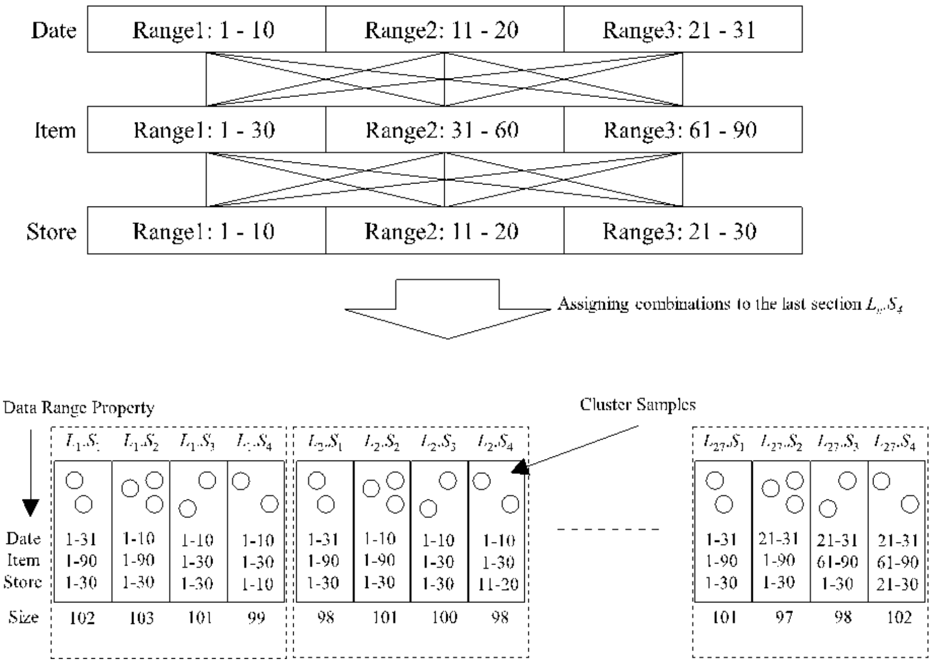


Fig. 1. An example of cluster samples.

The Multidimensional Cluster Sampling View can be simply implemented into an index table and a cluster table, as shown in Fig. 2. The area surrounded by dash lines in Fig. 2. represents a Cluster Sample. The index table contains all properties of the Cluster Samples. The cluster table contains actual data. The leaf and section number are shared between two tables for the join operation. For efficient disk access, the list partitioning technique is adopted in the cluster table.

2.2. Query Processing in Multidimensional Cluster Sampling View

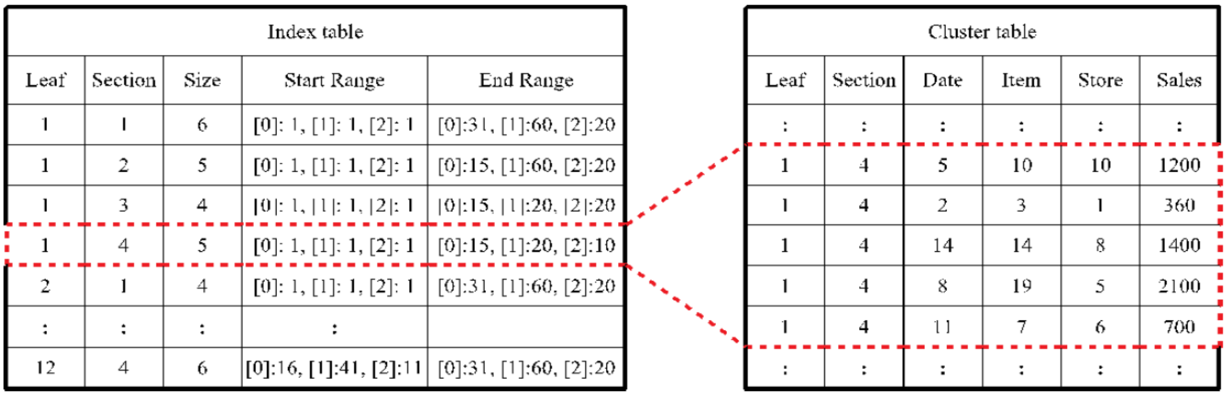
The purpose of the Multidimensional Cluster Sampling View is for efficient extraction of samples and effective approximation of query results. The conceptual process of query execution in the Multidimensional Cluster Sampling View can be described with the following example:

Let $Q = ([Date: 5 - 10], [Item: 20 - 40], [Store: 3])$ with 3.5% samples be the example query condition. The query for an average value with this condition can be specified with the following SQL-like query:

```
SELECT AVG(SALES_AMOUNT) FROM SALES SAMPLE (3.5%)
```

WHERE (DATE BETWEEN 5 AND 10)
AND (ITEM BETWEEN 20 AND 40)
AND (STORE = 3).

As shown in Fig. 3, there are three key attributes, namely the DATE attribute, ITEM attribute, and STORE attribute. The DATE attribute has three partitions, 1 to 10, 11 to 20, and 21 to 31, and the ITEM attribute has three partitions, 1 to 30, 31 to 60, and 61 to 90. The STORE attribute has three partitions, 1 to 10, 11 to 20, and 21 to 30. Therefore, the Cluster Samples have 27 leaves and 4 section types. In other words, there are 108 Cluster Samples. As a result, the size of one Cluster Sample is approximately 0.93% of the whole data.



Index table

Leaf	Section	Size	Start Range	End Range
1	1	6	[0]: 1, [1]: 1, [2]: 1	[0]:31, [1]:60, [2]:20
1	2	5	[0]: 1, [1]: 1, [2]: 1	[0]:15, [1]:60, [2]:20
1	3	4	[0]: 1, [1]: 1, [2]: 1	[0]:15, [1]:20, [2]:20
1	4	5	[0]: 1, [1]: 1, [2]: 1	[0]:15, [1]:20, [2]:10
2	1	4	[0]: 1, [1]: 1, [2]: 1	[0]:31, [1]:60, [2]:20
:	:	:	:	
12	4	6	[0]:16, [1]:41, [2]:11	[0]:31, [1]:60, [2]:20

Cluster table

Leaf	Section	Date	Item	Store	Sales
:	:	:	:	:	:
1	4	5	10	10	1200
1	4	2	3	1	360
1	4	14	14	8	1400
1	4	8	19	5	2100
1	4	11	7	6	700
:	:	:	:	:	:

Fig. 2. Mapping the multidimensional cluster sampling view.

Firstly, this method of processing filters the data range property of Cluster Samples by the query condition. In this example, seeing as the data range properties of all sections (S_1, S_2, S_3, S_4) in leaf L_1 satisfy the query condition, the processing method is used to extract the Cluster Samples (L_1S_1, L_1S_2, L_1S_3, L_1S_4). In leaf L_2, the three sections (S_1, S_2, S_3) are extracted for the same reason. Thus, the data range properties of all Cluster Samples are checked in relation to the query condition, and then several Cluster Samples are extracted. As a point to be noted, if the approximate query processing is for the total numbers of records or total values, only section 1 can be used to obtain samples for the estimator, which will be explained in Section 2.3. Otherwise, all sections can be used as samples.

Secondly, the query processing rearranges the filtered Cluster Samples in descending order by section number (L_1S_4, L_4S_4, $L_1S_3, \ldots,$ $L_{27}S_1$). As mentioned previously, sample records in the larger section number belong to the more specific data range. Therefore, random samples in the larger section number which have been filtered by the query Q are more relevant to the query condition. This can help to avoid missing data that is relevant to the query conditions, even in highly selective queries.

Finally, the query processing draws necessary samples from the head of the sorted clusters by using the size property of these clusters. In this example, seeing as the sampling rate is at 3.5%, four Cluster Samples (L_1S_4, L_4S_4, L_1S_3, L_2S_3), approximately 3.7%, are drawn.

2.3. Estimator Using Random Samples

An estimate for the total value requires an average value and size, which are obtained from the sample. An estimate for the average value can be simply calculated from the sample. Regarding an estimate for the size, Chaudhuri and Mukeriee [17] proposed an unbiased estimator for the size based on simple random sampling without replacement. The estimators which are described here are used as calculation methods of approximate answers after the query processing presented in Section 2.2.

- Let N be the total number of records in a complete data

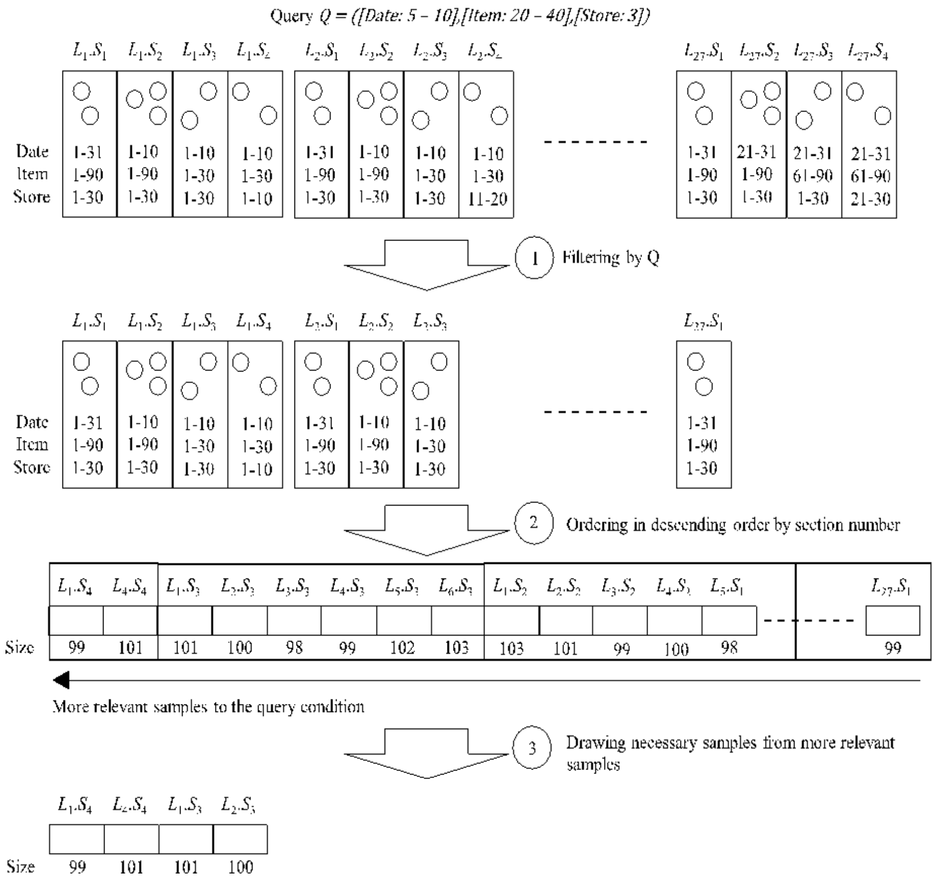

Query Q = ([Date: 5 – 10],[Item: 20 – 40],[Store: 3])
L1.S1 L1.S2 L1.S3 L1.S4 L2.S1 L2.S2 L2.S3 L2.S4 L27.S1 L27.S2 L27.S3 L27.S4
Date
Item
Store
1-31 1-90 1-30
1-10 1-90 1-30
1-10 1-30 1-30
1-10 1-30 1-10
1-31 1-90 1-30
1-10 1-90 1-30
1-10 1-30 1-30
1-10 1-30 11-20
1-31 1-90 1-30
21-31 1-90 1-30
21-31 61-90 1-30
21-31 61-90 21-30
1 Filtering by Q
L1.S1 L1.S2 L1.S3 L1.S4 L2.S1 L2.S2 L2.S3 L27.S1
2 Ordering in descending order by section number
L1.S4 L4.S4 L1.S3 L2.S3 L3.S3 L4.S3 L5.S3 L6.S3 L1.S2 L2.S2 L3.S2 L4.S1 L5.S1 L27.S1
Size 99 101 101 100 98 99 102 103 103 101 99 100 98 99
More relevant samples to the query condition
3 Drawing necessary samples from more relevant samples
L1.S4 L4.S4 L1.S3 L2.S3
Size 99 101 101 100

Fig. 3. Example query execution process using the multidimensional cluster sampling view.

- Let m be the number of sampled records satisfying the query conditions
- Let n_1 be the number of sampled records in S_1
- Let m_1 be the number of sampled records satisfying the query conditions in S_1

An unbiased estimator for the number of records is given by

$$\widehat{M} = N\frac{(m_1 - 1)}{(n_1 - 1)}. \tag{1}$$

An unbiased estimator for the average value is given by

$$\bar{x} = \frac{\sum_{k=0}^{m} Value_k}{m}. \tag{2}$$

An unbiased estimator for the total value is given by

$$\widehat{T} = \widehat{M}\,\frac{\sum_{k=0}^{m_1} Value_k}{m_1}. \tag{3}$$

2.4. Database Relevancy Ratio

Database Relevancy Ratio (DRR), which is advocated by Rudra, et al. [8], is a measure designed to represent query selectivity in order to evaluate accuracy of estimation of query results. The DRR is the ratio of the total number of records satisfying a query condition within a data set, which is denoted by $\rho(Q)$ with a query Q. For example, the DRR $\rho(Q)$ is 1 if there is no query condition. The DRR $\rho(Q)$ is 0.2 if the query condition narrows down the complete data to 20 percent.

3. Dimensionality Evaluation for Accuracy

3.1. Evaluation Approach

The objective of this section is to evaluate whether the number of dimensions affects the level of accuracy of approximation provided by the Multidimensional Cluster Sampling View. In terms of the effects of dimensionality on accuracy, some experiments were conducted in the Multidimensional Cluster Sampling View with 6, 10 and 13 dimensional key attributes. Detailed information regarding an original table of this view will be presented in Section 3.2. To evaluate the level of accuracy of this view, four DRR values were used: medium DRR (0.05), low DRR (0.01), very low DRR (0.001) and extremely low DRR (0.0001). The approximation was also compared with the absolute answer using three aggregate functions (AVG, COUNT and SUM), with the sampling rate ranging from 1% to 10%. The queries were repeated ten times on each sampling rate, and the mean value of the results was adopted as the approximate answer. Finally, the effect on accuracy of the number of dimensions is discussed in terms of error rates, which are calculated by the difference between the absolute and approximate answers expressed as a percentage of the absolute answers.

Table 1. Column Definitions of WEB_SALES Table and Target Key Attributes Used in Experiments for Effects of Dimensionality on Accuracy

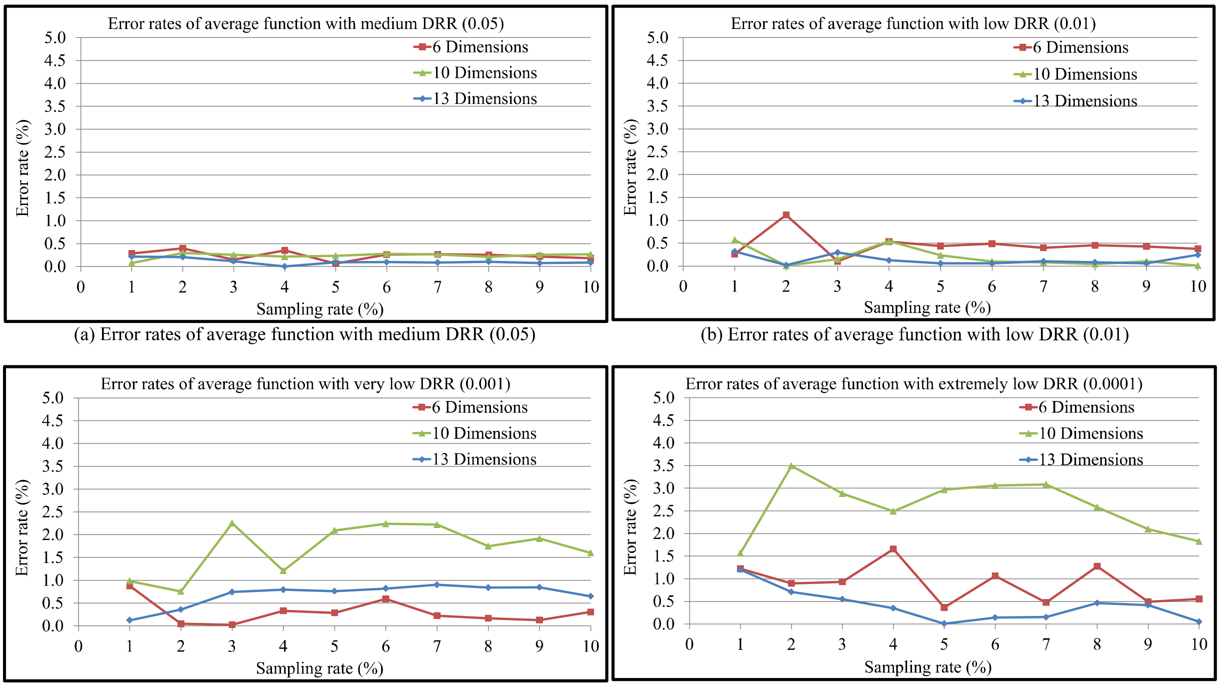
Column	Dimensional Table	6 Dimensions	10 Dimensions	13 Dimensions
ws_sold_date_sk	date_dim	✓	✓	✓
ws_sold_time_sk	time_dim	✓	✓	✓
ws_ship_date_sk	date_dim		✓	✓
ws_item_sk	item, web_returns	✓	✓	✓
ws_bill_customer_sk	customer	✓	✓	✓
ws_bill_cdemo_sk	customer_demographics		✓	✓
ws_bill_hdemo_sk	household_demographics		✓	✓
ws_bill_addr_sk	customer_address		✓	✓
ws_ship_customer_sk	customer			✓
ws_ship_cdemo_sk	customer_demographics			✓
ws_ship_hdemo_sk	household_demographics			✓
ws_ship_addr_sk	customer_address			✓
ws_web_page_sk	web_page	✓	✓	✓
ws_web_site_sk	web_site	✓	✓	
ws_ship_mode_sk	ship_mode			
ws_warehouse_sk	warehouse			
ws_promo_sk	promotion			
ws_order_number	web_returns			
ws_quantity				
ws_wholesale_cost				
ws_list_price				
ws_sales_price				
ws_ext_discount_amt				
ws_ext_sales_price				
ws_ext_wholesale_cost				
ws_ext_list_price				
ws_ext_tax				
ws_coupon_amt				
ws_ext_ship_cost				
ws_net_paid				
ws_net_paid_inc_tax				
ws_net_paid_inc_ship				
ws_net_paid_inc_ship_tax				
ws_net_profit				

3.2. Data Sets

The dataset is generated by *dsdgen*, the software package of TPC-DS [18]. The Multidimensional Cluster Sampling View was built from a fact table using the utility that was developed. The table named *WEB_SALES* is composed of approximately 72 million records. Table 1 depicts column definitions of the *WEB_SALES* table and the target key attributes used in the experiments. There are 18 foreign keys in the table, and some of them are used as dimensions in each experiment. In addition to this, a measure attribute named *WS_WXT_SALES_PRICE*, which ranges 0 to 29,810, was employed in the experiments.

3.3. Experimental Results and Analysis

In conducting the analysis, the experiments were mainly divided into three sets in terms of different aggregate functions: AVERAGE, COUNT and SUM function.



(a) Error rates of average function with medium DRR (0.05)

(b) Error rates of average function with low DRR (0.01)

(c) Error rates of average function with very low DRR (0.001)

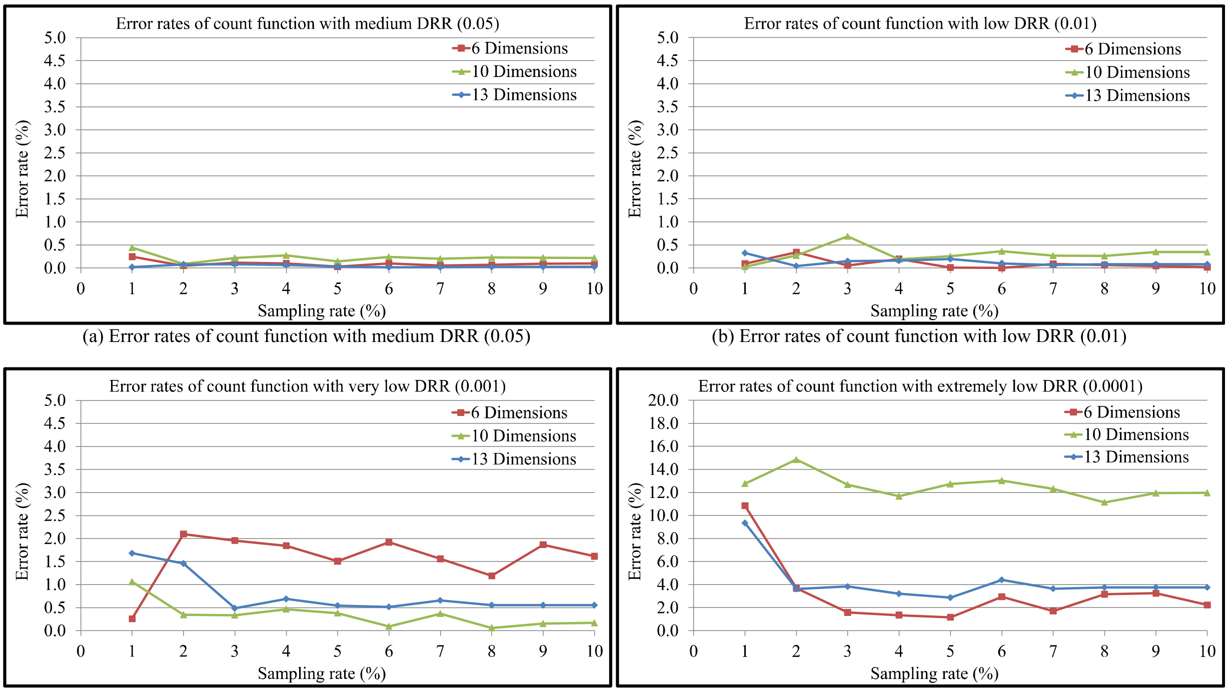
(d) Error rates of average function with extremely low DRR (0.0001)

Fig. 4. Error rates of AVERAGE function between absolute answers and approximate answers using the multidimensional cluster sampling view with various DRR values on three types of dimensions from 1% to 10% sampling rate.

For the first experiment, an examination was conducted regarding the effects of dimensionality on the accuracy of the AVERAGE function in the Multidimensional Cluster Sampling View. Fig. 4 includes four line graphs that demonstrate the error rates of AVERAGE function between the absolute and approximate answers obtained through the use of the Multidimensional Cluster Sampling View with various DRR values on three types of dimensions which ranged from a sampling rate of 1% to 10%. As shown in Fig. 4a, all results indicate very low error rates, all below 0.5% error. In this chart, no significant differences in accuracy are seen due to the effects of dimensionality. In Fig. 4b, all results except for the sampling rate of two percent on six dimensions show error rates below 0.5%. In Fig. 4c, whilst the error rates always remain below 2.5%, it is clearly evident that error rates vary for different numbers of dimensions. The results seen when there were 10 dimensions show the largest error rate of all, and then 13 and 6 dimensions follow suit respectively. Therefore, the dimensionality of queries using the Multidimensional Cluster Sampling View does not correlate with accuracy in this chart. This is due to there being no proportional relationship between the number of dimensions and error rates. Fig. 4d shows error rates below 3.5% in all results.

Similarly, for very low DRR (0.001), it can be clearly seen that there are differences among the three sets of dimensions. The largest error rate among them is seen when 10 dimensions are used. 6 dimensions and 13 dimensions follow respectively. Consequently, the effects of dimensionality on the accuracy of AVERAGE function using the Multidimensional Cluster Sampling View are not evident in the results of this experiment.

For the next experiment, the effects of dimensionality on the accuracy of COUNT function using the Multidimensional Cluster Sampling View were examined. Fig. 5 includes four line graphs which show the error rates of COUNT function between absolute and approximate answers obtained through the use of the Multidimensional Cluster Sampling View, with various DRR values on three types of dimensions ranging from sampling rates between 1% and 10%. As shown in Fig. 5a, all results indicate an error rate under 0.5%. No significant differences in accuracy are seen in this graph that could be attributed to the effects of dimensionality. Next, in Fig. 5b, all error rates except for the 3% sampling show below 0.5%. In Fig. 5c, all results remain low error rates, below 2.5%. It is clearly seen that error rates vary for different numbers of dimensions. The largest error rate is seen on 6 dimensions, followed in order by 13 dimensions and 10 dimensions. Therefore, the dimensionality of queries using the Multidimensional Cluster Sampling View does not correlate with accuracy in this graph. In Fig. 5d, error rates on ten dimensions show an approximate percentage of between 11 % and 15%. These error rates are clearly unacceptable for use in approximation. The approximation with 1% samples on 6 and 13 dimensions also show high error rates, approximately 11% and 9% respectively. In contrast, the error rates between 2% and 10% samples on 6 and 13 dimensions are acceptable, under 5%. Hence, the effects of dimensionality on the accuracy of the COUNT function using the Multidimensional Cluster Sampling View are not obvious in terms of this experiment.



(a) Error rates of count function with medium DRR (0.05)

(b) Error rates of count function with low DRR (0.01)

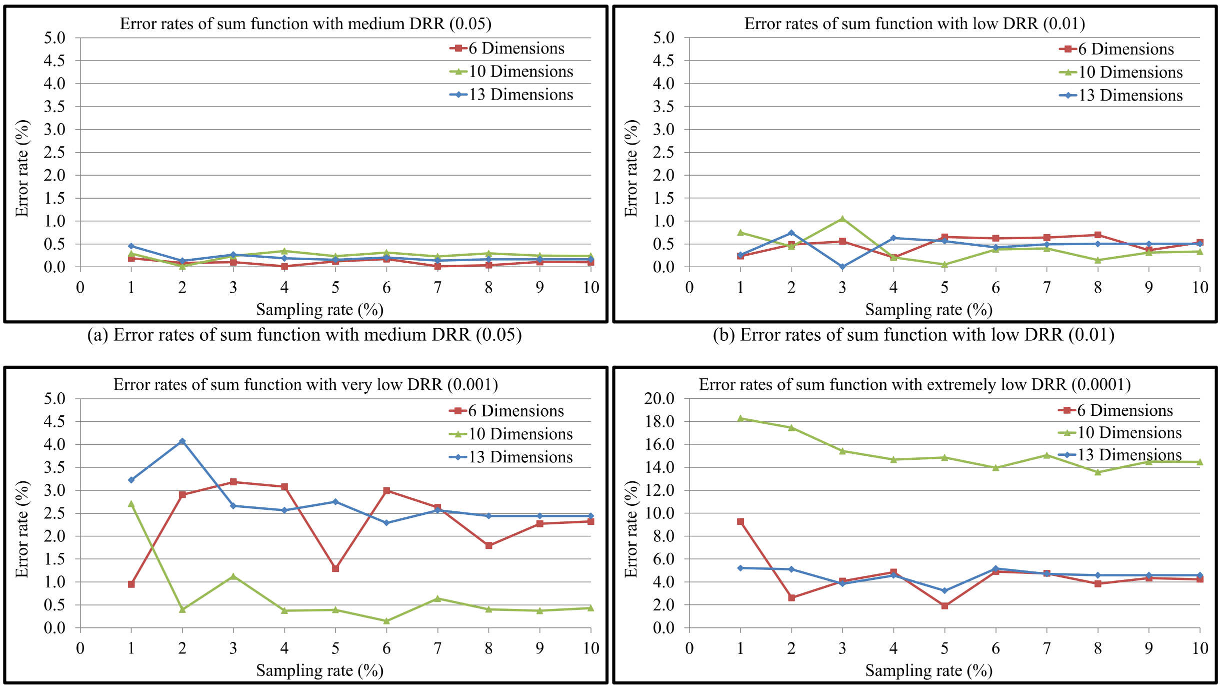
(c) Error rates of count function with very low DRR (0.001)

(d) Error rates of count function with extremely low DRR (0.0001)

Fig. 5. Error rates of COUNT function between absolute answers and approximate answers using the Multidimensional Cluster Sampling View with various DRR values on three types of dimensions from 1% to 10% sampling rate.

For the final experiment, the effects of dimensionality on the accuracy of the SUM function using the

Multidimensional Cluster Sampling View were examined. Fig. 6 includes four line graphs that demonstrate the error rates of SUM function between absolute and approximate answers obtained through the use of the Multidimensional Cluster Sampling View, with various DRR values on three types of dimensions ranging from sampling rates between 1% and 10%. As shown in Fig. 6a and Fig. 6b, all results maintain low error rates, under 1%. The significant effects of dimensionality on accuracy are not visible in these charts. In Fig. 6c, error rates on 6 dimensions and 13 dimensions are relatively large on the whole, but below 4%. The error rates on 10 dimensions show the lowest values except for the sampling rate of 1%. In this graph, the effects of dimensionality on accuracy are also unseen. In Fig. 6d, the results on 10 dimensions show the largest error, between 13% and 18%. The error rates on 6 and 13 dimensions remain almost similar on the whole, around 5%. No significant effects of dimensionality on the accuracy are seen in this graph. Therefore, the number of dimensions does not correlate with the accuracy of SUM function using the Multidimensional Cluster Sampling View.



(a) Error rates of sum function with medium DRR (0.05)

(b) Error rates of sum function with low DRR (0.01)

(c) Error rates of sum function with very low DRR (0.001)

(d) Error rates of sum function with extremely low DRR (0.0001)

Fig. 6. Error rates of SUM function between absolute answers and approximate answers using the Multidimensional Cluster Sampling View with various DRR values on three types of dimensions from 1% to 10% sampling rate.

Thus, there are three findings that can be reached from these experiments. Firstly, the number of dimensions in the Multidimensional Cluster Sampling View does not affect the accuracy of the approximation. Secondly, the approximation of the AVERAGE function using the Multidimensional Cluster Sampling View could provide a reliable level of accuracy even in extremely selective queries (0.0001 DRR). Finally, the approximation of the AVERAGE function using the Multidimensional Cluster Sampling View could provide a more accurate result than that provided for the COUNT or SUM functions. In the experiments conducted, the approximation of the COUNT or SUM functions using this view could provide an acceptable level of accuracy within a very low DRR (0.001 DRR), whereas it could not provide an acceptable level of accuracy within an extremely low DRR (0.0001 DRR). This is because Cluster Samples with larger section numbers, which belong to specific data ranges, cannot be used as samples of the total number of records or the total size. Therefore, the approximation using this view is more reliable for mean values than for the number of records or total values.

For the additional evaluation, an experiment was conducted regarding the effects of sample size on the accuracy in the Multidimensional Cluster Sampling View. Fig. 7 shows error rates of COUNT function using the Multidimensional Cluster Sampling View with 0.0004 DRR ranging from sampling rate between 1% and 25%. It is clearly seen that accuracy of the approximate answers in this view tends to improve with increasing the sample size.

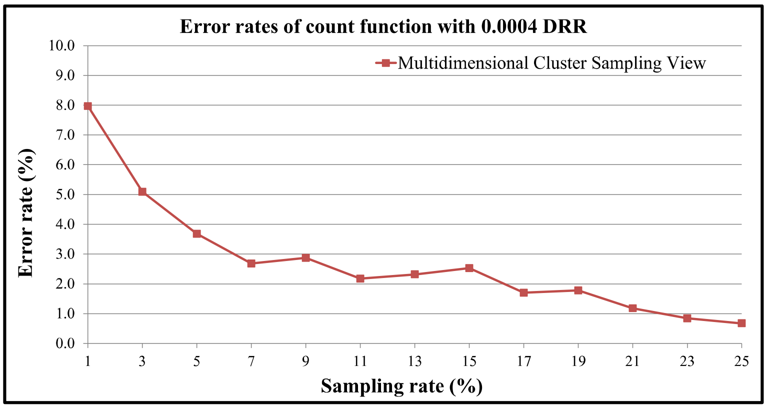


Fig. 7. Error rates of COUNT function between absolute answers and approximate answers using the multidimensional cluster sampling view with 0.0004 DRR from 1% to 25% sampling rates.

4. Dimensionality Evaluation for Construction Performance

4.1. Evaluation Approach

The aim of this section is to evaluate whether dimensionality affects the construction performance of the Multidimensional Cluster Sampling View. In order to compare the effects of dimensionality on construction time, the performance between one and ten dimensions was measured using the utility developed. Detailed information regarding an original table of the view will be presented in Section 4.2. All experiments were carried out in CentOS 6.5 (64 bit) with Oracle database 11g Release 2 on VMware Player 6.1, having 4 Gigabytes of RAM, four 3.4 GHz clock processors and a 1 Terabyte disk.

Table 2. Target Key Attributes used in Experiments for Effects of Dimensionality on Construction Performance

No.	Column Name	Dimensions
1	CS_SOLD_DATE_SK	✓
2	CS_SOLD_TIME_SK	✓
3	CS_SHIP_DATE_SK	✓
4	CS_BILL_CUSTOMER_SK	✓
5	CS_BILL_CDEMO_SK	✓
6	CS_BILL_HDEMO_SK	✓
7	CS_BILL_ADDR_SK	✓
8	CS_SHIP_CUSTOMER_SK	✓
9	CS_SHIP_CDEMO_SK	✓
10	CS_SHIP_HDEMO_SK	✓

4.2. Data Sets

The raw data is populated by the data generator of TPC-DS using a scale factor of 1GB [18]. A table named *CATALOG_SALES* is used to build the Multidimensional Cluster Sampling View with a variety of dimensions. The table is composed of 34 columns with approximately 1.45 million records. Table 2 shows the target key attributes used in this experiment. For example, the top three attributes, *CS_SOLD_DATE_SK*, *CS_SOLD_TIME_SK* and *CS_SHIP_DATE_SK*, are used as key attributes to measure the construction time in regards to the three dimensions.

4.3. Experimental Results and Analysis

Fig. 8 shows the construction time of the Multidimensional Cluster Sampling View between one and ten dimensions. Between one and seven dimensions, the performance shows around 2.5 minutes. The effects of dimensionality on the construction time cannot be clearly seen in regards to these dimensions. However, it shows 5 minutes on eight dimensions, approximately double when compared with seven dimensions. Furthermore, it shows 12.5 minutes (five times) and 18.8 minutes (seven and a half times) on nine and ten dimensions respectively. Therefore, it appears that there is another factor which could affect the construction time.

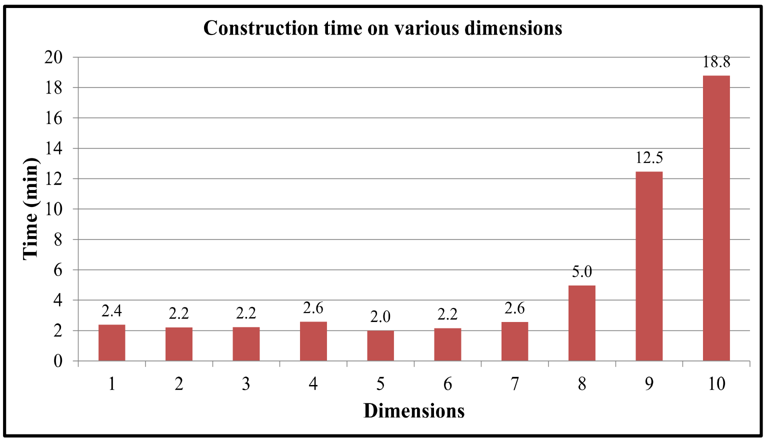


Fig. 8. Construction time of the multidimensional cluster sampling View between one and ten dimensions.

Table 3 shows the number of Cluster Samples on each dimension in Fig. 8. Recall that Cluster Samples are containers to divide a data set into many groups at random with some given properties as described in Section 2.1. As shown in the table, the number of Cluster Samples between one and seven dimensions, with the exception of 6 dimensions, is around 1,000. Then, the number on eight dimensions and nine dimensions is double and five times respectively. This rate of increase is the same as that of construction time. It seems that the number of Cluster Samples could affect the construction time.

Table 3. The Number of Cluster Samples on Each Dimension in Fig. 8

Dimensions	The number of Cluster Samples
1	1,000
2	972
3	864
4	960
5	972
6	672
7	1,024
8	2,304
9	5,120
10	11,264

Consequently, a further experiment was conducted to confirm whether the number of Cluster Samples has an impact on the construction time. The performance and Cluster Sample size was measured on four fixed dimensions. As shown in Fig. 9, 2.1 minutes are required for the construction with 960 Cluster

Samples, and it takes 3.4 minutes for the construction with 2,000 Cluster Samples. It takes 4.6 minutes for the construction with 3,125 Cluster Samples. As expected, the construction time increases almost in direct proportion to the number of Cluster Samples.

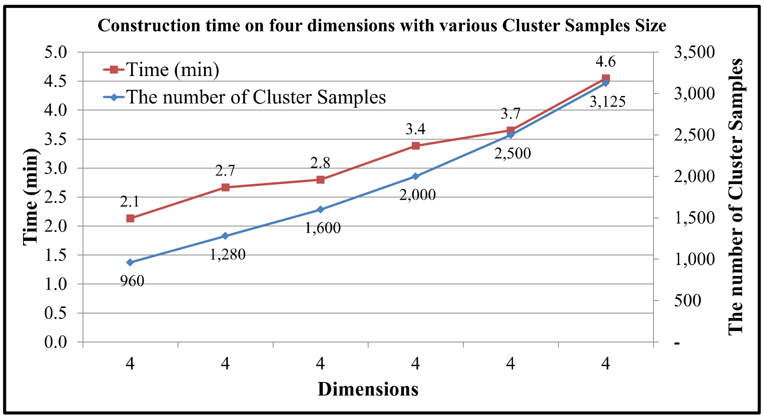


Fig. 9. Construction time on four dimensions between 960 and 3,125 cluster samples.

In summary, the factor which has an impact on the construction time of the Multidimensional Cluster Sampling View is not the dimensionality, but rather the Cluster Sample size. In a strict sense, although the dimensionality or other factors might affect the construction time, the effects could be minimal.

5. Conclusion

In this paper, the effects of dimensionality on both the level of accuracy of approximation and construction time were examined using the Multidimensional Cluster Sampling View. It was discovered that the level of accuracy of approximation using this view was not affected by the number of dimensions. Also, it was determined that the number of dimensions did not affect the construction time of the view, but the Cluster Sample size affected the construction performance. Hence, the Multidimensional Cluster Sampling View is capable even in high-dimensional data if the number of Cluster Samples is limited to a certain number.

Future research may investigate the suitable size of Cluster Samples for the best performance and evaluate further practicality of the Multidimensional Cluster Sampling View, such as in terms of using various real-life data sets.

References

[1] Chakrabarti, K., Garofalakis, M., Rastogi, R., & Shim, K. (2001). Approximate query processing using wavelets. *The VLDB Journal, 10*, 199-223.

[2] Garofalakis, M. N., & Gibbon, P. B. (2001). Approximate query processing: Taming the terabytes (tutorial). *Proceedings of the 27th International Conference on very Large Data Bases*.

[3] Babcock, B., Chaudhuri, S., & Das, G. (2003). Dynamic sample selection for approximate query processing. *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*.

[4] Ruoming, J., Glimcher, L., Jermaine, C., & Agrawal, G. (2006). New sampling-based estimators for OLAP queries. *Proceedings of the 22nd International Conference on Data Engineering* (pp. 18-18).

[5] Chaudhuri, S., Das, G., & Narasayya, V. (2007). Optimized stratified sampling for approximate query processing. *ACM Trans. Database Syst., 32*.

[6] Liu, Q. (2009). Approximate query processing. *Encyclopedia of Database Systems*, 113-119.

[7] Wang, Y. J., Wang, H. H., & Li, H. (2013). A histogram based analytical approximate query processing for

massive data. *Applied Mechanics and Materials*.

[8] Rudra, A., Gopalan, R., & Achuthan, N. R. (2012). An efficient sampling scheme for approximate processing of decision support queries. presented at t the 14th International Conference on Enterprise Information Systems, Wroclaw, Poland.

[9] Olken, F., & Rotem, D. (1986). Simple random sampling from relational databases. *Proceedings of the 12th International Conference on Very Large Data Bases*.

[10] Olken, F., & Rotem, D. (1995). Random sampling from databases: A survey. *Statistics and Computing*.

[11] Joshi, S., & Jermaine, C. (2008). Robust stratified sampling plans for low selectivity queries. *Proceedings of the IEEE 24th International Conference on Data Engineering* (pp. 199-208).

[12] Joshi, S., & Jermaine, C. (2008). Materialized sample views for database approximation. *IEEE Transactions on Knowledge and Data Engineering*, *20*, 337-351.

[13] Li, X., Han, J., Yin, Z., Lee, J. G., & Sun, Y. (2008). Sampling cube: A framework for statistical olap over sampling data. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*.

[14] Webb, L. M., & Wang, Y. (2014). Techniques for sampling online text-based data sets. *Big Data Management, Technologies, and Applications*.

[15] Inoue, T., Krishna, A., & Gopalan, R. (2015). Multidimensional cluster sampling view on large databases for approximate query processing. *Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference (EDOC)*.

[16] Kimball, R., Ross, M., & Thornthwaite, W. (2011). *Data Warehouse Lifecycle Toolkit*.

[17] Chaudhuri, A., & Mukeriee, B. (1985). Domain estimation in finite populations. *Australian Journal of Statistics, 27*, 135-137,.

[18] TPC. (2014). TPC BENCHMARK ™ DS Standard Specification Version 1.1.0. Retrieved April 30, 2004, from http://www.tpc.org/tpcds/

Tomohiro Inoue is an MPhil candidate in computer science at Curtin University, Australia. He received a BSc degree in mathematics from Shizuoka University, Japan. He also worked as a software engineer for NEC Corporation in Japan for 14 years. His research interests include databases, big data analytics, and approximation.

Aneesh Krishna is currently senior lecturer of software engineering with the Department of Computing, Curtin University, Australia. He holds a PhD in computer science from the University of Wollongong, Australia, an M.Sc. (Eng.) in electronics engineering from Aligarh Muslim University, India and a B.E. degree in electronics engineering from Bangalore University, India. He was a lecturer in software engineering at the School of Computer Science and Software Engineering, University of Wollongong, Australia (from February 2006 - June 2009). His research interests include software engineering, requirements engineering, conceptual modelling, agent systems, formal methods and service-oriented computing. His research is (or has been) funded by the Australian Research Council and various Australian government agencies as well as companies such as Woodside Energy, Amristar Solutions, Andrew Corporation, NSW State Emergency Service, Western Australia Dementia Study Centre and Autism West. He serves as assessor (Ozreader) for the Australian Research Council. He has been on the organising committee, served as invited technical program committee member of many conferences and workshop in the areas related to his research.

Raj P. Gopalan is currently an adjunct senior research fellow in the Department of Computing, Curtin University, Australia. He has a PhD in computer science from the University of Western Australia, a MSc by research in computer science from the National University of Singapore, a post graduate diploma in management from the Indian Institute of Science, Bangalore and a BSc in Mechanical Engineering from Kerala University. His research interests include database systems, data mining and data-driven software engineering.