**User Manual**

1. **Procedure to obtain the output for Fig. 5a**

**Input:**

1. First File namely **sequentialdatabase.txt** represent the dataset

- It consist of transactions namely 3 15 7 10, where 3 represent total number of items in the transaction and 15, 7, 10 represent items

1. Second File namely **seq(item).txt** represent the distinct items such as 1to 30
2. Third File namely **seq(txn).txt,** which consist of total number of transaction in incremental and existing dataset

-We need to **modify the content of seq(txn).txt** file while size of incremental dataset varies.

1. The minimum support =0.01 and total number of partition=2 are entered in the console.

To obtain the output for Fig. 5a, we need to create the above three files namely sequentialdatabase.txt, seq(item).txt, seq(txn).txtand run the program as follows

1. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the **10%** of incremental dataset
2. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **160000 and 200000,** which give the running time for the **20%** of incremental dataset
3. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **120000 and 200000,** which give the running time for the **40%** of incremental dataset
4. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **80000 and 200000,** which give the running time for the **60%** of incremental dataset
5. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **40000 and 200000,** which give the running time for the **80%** of incremental dataset
6. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **0 and 200000,** which give the running time for the **100%** of incremental dataset

The runtime of RePLNI algorithm for the min\_sup = 0.01, no of partition = 2 and 20% of incremental dataset is pictorially represented in the screenshot, which shown in Fig. 1.

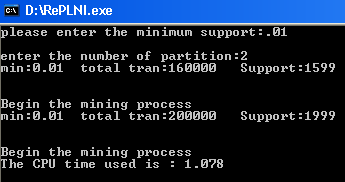


Fig. 1. Screenshot for incremental mining in RePLNI algorithm

**Running Time for RePLNI and PLWAP**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| I/II | **180K/200K** | **160K/200K** | **120K/200K** | **80K/200K** | **40K/200K** | **0/200K** |
| **RePLNI** | .906 | 1.078 | 1.453 | 1.781 | 2.047 | 2.344 |
| **PLWAP(I)/(II)** | 2.219/2.437 | 2/2.437 | 1.484/2.437 | 0.985/2.437 | 0.531/2.437 | -/2.437 |

Since, the PLWAP will not support incremental mining it will be executed twice. That is first for mining in the initial transactions, for e.g., if we consider 20% of U then the algorithm run for 160k (20% of U) transaction and second time for both the initial (160000) and incremental dataset (40000), it mean for whole dataset, which is shown in Fig.2.

|  |  |
| --- | --- |
| (a) | (b) |

Fig. 2. Screenshot for incremental mining in PLWAP algorithm

We not get the third party code of incremental algorithm (RePL4UP) however we have third party code of non incremental algorithm (PLWAP). Hence, in-order to prove our proposed algorithm, we calculated the relative execution time which described clearly in the following passage.

**Calculating Relative Execution Time**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **10%** | **20%** | **40%** | **60%** | **80%** | **100%** |
| **RePL4UP/PLWAP** | 6.991 | **4.083** | **1.976** | **1.496** | **1.101** | **1.248** |
| **RePLNI/PLWAP (max)** | 5.139 | **4.319** | **3.204** | **2.614** | **2.275** | **1.986** |
| **RePLNI/PLWAP** | 5.139 | **4.115** | **2.699** | **1.9239** | **1.450** | **1.040** |

The relative execution time for 10% of 200K as given above is not encouraging to the proposed algorithm. This is clearly stated in the submitted paper **(in page number: 15 and line number: 2)**. The relative execution time for RePL4UP/PLWAP is obtained from (**Ezeife et al, 2009**, **page number: 408 and in Table 5**) by dividing runtime of PLWAP/ runtime of RePL4UP. For e.g., let us consider for 20% is 3.062/.75=**4.083**. The relative execution time for RePLNI/PLWAP is obtained by dividing (PLWAP(I)+(II))/RePLNI for e.g., 20% of 200K => (2+2.437)/1.078 => 4.437/1.078=>**4.115.**

The relative execution time for RePLNI/PLWAP(max) is obtained by dividing max(PLWAP(I)+(II))/ RePLNI, For e.g., 20% of 200K=>4.656/1.078=>**4.319**, since, in the paper (**Ezeife et al, 2009**, **page number: 408, Table 5 and PLWAP algorithm**) the value of PLWAP algorithm in Table 5 is constant 3.062. Hence, we also obtain the constant value of PLWAP by finding max (PLWAP(I)+(II)) = 4.656, which divided with all the values in RePLNI to obtain RePLNI/PLWAP (max). However, in both way of calculation, such as RePLNI/PLWAP (max) and RePLNI/PLWAP, our evaluation is performed well compared to RePL4UP/PLWAP. Fig. 5a is plotted for **RePLNI/PLWAP (max) and RePL4UP/PLWAP.**

1. **Procedure to obtain the output for Fig. 5b**

**Input:**

We use

1. Files namely **sequentialdatabase.txt, seq(item).txt, seq(txn).txt.**
2. The min supports and total number of partition are entered in the console.

To obtain the output for Fig. 5b, we need to create the above three files and run the program as follows

1. Enter min\_sup=0**.001**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the support **.001** of 10% of incremental dataset
2. Enter min\_sup=**0.005**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the support **.005** of 10% of incremental dataset
3. Enter min\_sup=**0.01**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the support **.01** of 10% of incremental dataset
4. Enter min\_sup=**0.05**, no of partition: **2** and seq(txn).txt contain the values namely 1**80000 and 200000,** which give the running time for the support **.05** of 10% of incremental dataset
5. Enter min\_sup=**0.1**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the support **.1** of 10% of incremental dataset
6. Enter min\_sup=**0.15**, no of partition: **2** and seq(txn).txt contain the values namely **180000 and 200000,** which give the running time for the support **.15** of 10% of incremental dataset

**Running Time for RePLNI and PLWAP**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **0.005** | **0.01** | **0.05** | **0.1** | **0.15** | **0.2** |
| **RePLNI** | 1.422 | .938 | .359 | .235 | .219 | .203 |
| **PLWAP(180K) (I)** | 2.594 | 2.218 | 1.578 | 1.422 | 1.391 | 1.36 |
| **PLWAP(200K) (II)** | 2.86 | 2.438 | 1.781 | 1.562 | 1.563 | 1.5 |
| **(I+II)** | 5.454 | 4.76 | 3.359 | 2.984 | 2.954 | 2.86 |

PLWAP algorithm is executed for two times. For first time, it is executed for dataset size of 180K by varying the support values. Second time, it is executed for incremental dataset i.e., 10% of U (20000 txn), which mean for entire dataset. The runtime for RePLNI and PLWAP algorithms is calculated as shown in Fig. 1 and Fig. 2 respectively by varying support and the size of both incremental and updated dataset is fixed.

**Calculating Relative Execution Time**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **0.005** | **0.01** | **0.05** | **0.1** | **0.15** | **0.2** |
| **RePL4UP/PLWAP** | 4.727 | 6.405 | **9.2** | **5.786** | **5.373** | **5.744** |
| **RePLNI/PLWAP** | 3.835 | 5.075 | **9.356** | **12.697** | **13.48** | **14.088** |

The relative execution times of the min\_sup 0.005 and 0.01 are not encouraging, the reasons are mentioned in the submitted paper **(in page number: 16 and line number: 19)**. Similar to earlier case, the relative execution time for **RePL4UP/PLWAP** is obtained from (**Ezeife et al, 2009**, **page number: 405 and in Table 3**). The relative execution time of **RePLNI/PLWAP** isobtained by dividing (RePLNI/(I+II))

1. **Procedure to obtain the output for Fig. 5c**

The relative execution times for 0.5%, 1% and 5% minsup are obtained by following the procedure explained in section A. Only the min\_sup values varied with 0.005, 0.01 and 0.05 respectively.

1. **Procedure to obtain the output for Fig. 5d**

The relative execution times for 40% of U, 60% of U and 80% of U are obtained by following the procedure explained in section B. Only content of seq(txn).txt is varied as given below

1. In case of **40% of U**, it contain **1,20, 000** and **200000**
2. In case of **60% of U**, it contain **80, 000** and **200000**
3. In case of **80% of U**, it contain  **40, 000** and **200000**
4. **Procedure to obtain the output for Fig. 5e**

Here, we are using runtime of both the proposed work and PLWAP algorithm, since RePL4UP algorithm will not support interactive mining.

**Input:**

We use

1. Files namely **sequentialdatabase.txt, seq(item).txt.**
2. In console, the min supports and total number of transactions are entered.

To obtain the output for Fig. 5e, we need to create the above two files and enter the value of total number of transactions as 200K and min\_sup as values given in x-axis of Fig. 5e. In order to continue for the next min\_sup, we have to give ‘0’ otherwise ‘1’, which is shown Fig. 3.

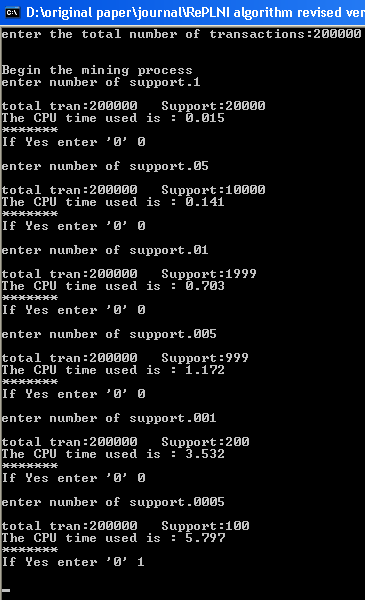


Fig. 3. Screenshot for interactive mining in RePLNI algorithm