ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Indian Journal of Science and Technology, *Vol 9(10), DOI: 10.17485/ijst/2016/v9i10/87954, March 2016*

Information Retrieval using Dynamic Decision Quadtree in Soil Database

A. Meenakshi[1*], P. Suganthi[2], R. Aghila[2] and S. Nirmala[2]

[1]Department of Information Technology, K.L.N. College of Information Technology, Madurai – 630612, Tamil Nadu, India; rajeshmeenakshi1@gmail.com

[2]Department of Computer Science and Engineering, K.L.N. College of Information Technology, Madurai – 630612, Tamil Nadu, India; suga.pathma@gmail.com, aghila25481@rediffmail.com, nirmalasme@gmail.com

Abstract

Background: Data volume with respect to modern systems has been growing outwardly at a rapid rate. The systems also face a tough task of frequent re-scanning of the large datasets stored because of the updation process. Frequent re-scanning, updation and large corpus data leads to having larger retrieval time and decreased efficiency. Hence, the data retrieved by automated systems may not be efficient and accurate. Knowledge Based System using Dynamic decision Quad tree is employed in this paper. The Quad trees designed in this chapter helps in retrieving knowledge about the suitable plants for the given kind of soil. **Methods:** The Quad trees designed in this paper helps in retrieving knowledge about the suitable plants for the given kind of soil. The Dynamic decision Quad tree is built using the knowledge and information given by Edaphologists and domain experts. The system is made of two modules, namely Dynamic Quad Tree construction module and information retrieval module. In the first module, the obtained data is transformed to a dynamic Quad tree based knowledge data structure. In the information retrieval module, knowledge retrieval is carried out with the help of constructed knowledge base (XML). **Findings:** Efficient techniques have been developed and tailored for solving complex soil datasets using data mining. The proposed scheme exploits clustering and dynamic decision tree based system for better storage in edaphology compared to existing systems. Dynamic decision quad trees are tailored to make best retrieval in soil databases. **Applications:** Assist Edaphologists and agricultural experts in obtaining the right crops/plants for the given soil characteristics.

Keywords: Data mining, Edaphology, Information Retrieval, Knowledge Extraction

1. Introduction

Data volume with respect to modern systems has been growing outwardly at a rapid rate[1]. The systems also face a tough task of frequent re-scanning of the large datasets stored because of the updation process. Frequent re-scanning, updation and large corpus data leads to having larger retrieval time and decreased efficiency. Hence, the data retrieved by automated systems may not be efficient and accurate. Knowledge Based System using Dynamic decision Quad tree is employed in this paper. The Dynamic decision Quad trees designed in the paper helps in retrieving knowledge about the suitable plants for the given kind of soil[2, 3]. The Quad tree is built using the knowledge and information given by Edaphologists and domain experts[4, 5]. The system is made of two modules, namely Dynamic decision Quad Tree construction module and information retrieval module. In the Quad tree module, obtained data is transformed to a dynamic Quad tree based knowledge data structure[6]. In the information retrieval module, plant names are retrieved based on the input soil characteristics by the user. When the query is posted by the user with input parameters it fetches the appropriate plants from the quad tree and gives decision supporting information which helps Edaphologists and agricultural experts from identifying the right crops/plants for the given soil characteristics. In this paper, effective knowledge retrieval is carried out with the aid of XML architecture and XML

*Author for correspondence

tags. The system is designed for retrieving best plants for the input soil condition given by the user. The obtained results prove that the system has performed well by having good evaluation metric values.

A dynamic decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility[7,8]. They are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal[9,10].

Initially, the knowledge is stored in the relational database with the input from the Edaphologists[11]. Here, it comprises of two tables of which one contains the plant details and the other contains the soil description. The plant details table consists of plant names, geology and taxonomy corresponding to the plant ID. Figure 2 shows an example of plant table *P* having attributes plant identification number *I*, name *Na*, geology *Ge* and the taxonomy *Ta*. We can see that a plant can have multiple plant IDs and the geology and taxonomy vary accordingly. The description table contains the plant ID, depth and the description of the soil. It also has the values of various parameters like clay, silt, sand, Ph, electrical conductivity, Calcium, Magnesium, Sodium, Potassium and Phosphorus Pent-oxide, Potassium Oxide[12].

The soil characteristics for the plant ID changes with the depth and because of that, each plant ID has more than one soil characteristics attached to it[12]. Figure 3 gives an example of soil characteristics table *S* having attributes of plant identification number *I*, depth *D*, description *G*, clay *Cl*, silt *Sl*, sand *Sa*, hydrogen ion concentration *H*, electrical conductivity *E*, calcium *Ga*, magnesium *M*, sodium *Ns*, potassium *Pt*, phosphorous pent oxide *Ph* and potassium oxide *Po*.

The Data flow of the proposed work is given in Figure 4.The input information is transformed into clusters by using key values for every plant data. And also the list is prepared for same plant IDs. Once the clusters are formed then it is transformed into dynamic decision tree[13] as shown in Figure 1.

2. Clustering and Dynamic Decision Tree Construction

The combination of clustering and dynamic decision tree is used for storage as given in Table 1. By using serializable concept the soil characteristics given in Table 2 are stored in a binary text file. In Edaphology there are two modules for storing the data in the database. They are:

- Data preprocessing.
- Clustering and Dynamic decision tree construction.

2.1 Data Pre-Processing

Data preparation is the first step for converting the information obtained from Edaphologists to Quad tree format.

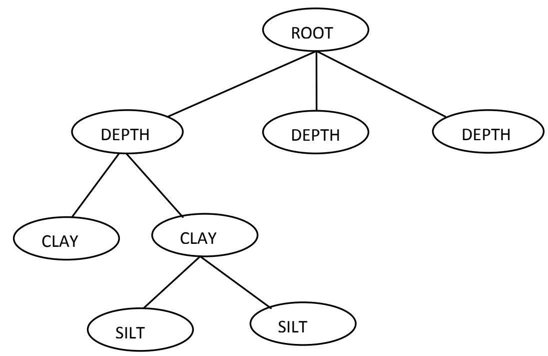

Figure 1. Sample dynamic decision tree.

Id	name	Geology	Taxonomy
0001	Prosophis juliflora, Cyprus sp., Hariyali,	Clay	Fine, montmorillonitic, isohyperthermic, noncalcareous, Chromic Haplusterts
0003	Palmyrah	Granite	Fine, mixed, isohyperthermic, noncalcareous, Typic Rhodustalfs
0017	Eucalyptus, Palmyrah, Neem, Tamarind	Laterite	Fine, mixed, isohyperthermic, noncalcareous, Typic Haplustepts
0019	Palmyrah, Neem	Granite	Clayey, mixed, isohyperthermic, noncalcareous, Lithic Haplustepts
0021	Palmyrah, Prosophis juliflora	Granite	Fine, mixed, isohyperthermic, noncalcareous, Typic Haplustalfs
0023	Neem, Palmyrah, Prosophis juliflora, Ta	Sand	Loamy-over-sandy, mixed, isohyperthermic, noncalcareous, Typic Ustifluvents
0024	Neem, Prosophis, Tamarind	Granite	Sandy, mixed, isohyperthermic, calcareous, Typic Ustorthents
0032	Palmyrah	Sand	Sandy, mixed, isohyperthermic, noncalcareous, Aquic Ustipsamments
0035	Neem, Palmyrah	Granite	Fine, mixed, isohyperthermic, calcareous, Calcic Haplustepts
0037	Neem, Prosophis juliflora	Western Ghats	Fine, mixed, isohyperthermic, noncalcareous, Typic Haplustepts
0039	Neem, Palmyrah, Tamarind	Granite	Fine, mixed, isohyperthermic, noncalcareous, Typic Haplustepts
0041	Palmyrah, Neem, Accacia	Granite	Clayey-skeletal, mixed, isohyperthermic, noncalcareous, Typic Haplustepts
0042	Ipomea, Thespesia populanea, Vagai	Clay	Fine, mixed, isohyperthermic, noncalcareous, Typic Rhodustalfs
0045	Palmyrah, Neem, Prosophis juliflora	Granite	Loamy, mixed, isohyperthermic, noncalcareous, Lithic Ustorthents
0047	Eucalyptus, Vagai	Laterite	Fine, mixed, isohyperthermic, noncalcareous, Fluventic Haplustepts
0050	Prosophis juliflora, Neem, Vetiver	Granite	Clayey-skeletal, mixed, isohyperthermic, noncalcareous, Lithic Ustorthents
0051	Prosophis juliflora, Palmyrah	Granite	Fine-loamy, mixed, isohyperthermic, noncalcareous, Fluventic Haplustepts
0055	Prosophis juliflora, Palmyrah, Tamarind	Granite	Fine-loamy, mixed, isohyperthermic, noncalcareous, Typic Haplustepts

Figure 2. The Plant table.

Id	Depth	Desc1	Clay	Silt	Sand	PH	EC	Ca	Mg	Na	K	P2O5	K2O
0001	0-13	Dark brown (10 YR 4/3); sandy clay; m	38.46	15.84	45.70	8.30	1.30	11.80	4.10	3.59	0.40	10.00	120.00
0001	13-65	Dark grayish brown (10 YR 4/2); sandy	41.92	18.07	40.01	8.90	1.50	13.87	4.30	4.07	0.46	10.00	80.00
0001	65-184	Very dark grayish brown (10 YR 3/2); c	40.84	21.90	37.26	8.90	1.40	14.20	3.90	4.09	1.02	0	0
0003	0-18	Red (2.5 YR 4/6); sandy clay loam; mo	24.60	15.2	60.2	7.36	0.04	8.0	1.5	0.68	0.32	29.45	228.00
0003	18-42	Red (2.5 YR 4/6); sandy clay loam; mo	25.60	16.8	57.60	7.11	0.04	10.0	2.0	0.73	0.35	27.37	376.00
0003	42-80	Dark yellowish brown (10 YR 4/6); cla	38.14	24.2	37.66	6.85	0.09	15.5	1.5	2.59	6.70	29.41	415.00
0003	80-140	Dark yellowish brown (10 YR 4/6); cla	44.60	25.6	29.80	6.81	0.08	18.5	1.5	4.25	4.62	15.60	425.00
0017	0-10	Dark yellowish brown (10 YR 4/4); sar	38.24	13.64	48.12	6.61	0.03	5.50	8.50	0.46	0.33	16.58	69.00
0017	10-29	Brownish yellow (10 YR 6/6); sandy cl	36.92	14.06	49.02	7.02	0.03	5.50	5.00	0.51	0.25	9.50	42.00
0017	29-57	Brownish yellow (10 YR 6/6); clay; mo	43.82	20.90	35.28	7.12	0.02	6.00	4.00	0.46	0.25	10.68	46.00
0017	57-87	Yellowish brown (10 YR 5/4); clay; mo	46.16	19.56	34.28	7.18	0.03	6.50	4.00	0.47	0.25	12.67	53.00
0017	87-1000	Weathered parent	0	0	0	0	0	0	0	0	0	0	0
0019	0-12	Very dark grayish brown (10 YR 3/2); c	44.60	20.24	35.16	8.19	0.07	31.00	6.50	0.36	0.28	20.78	122.00
0019	12-25	Very dark grayish brown (10 YR 3/2); g	48.40	21.26	30.34	8.33	0.05	32.00	8.00	0.38	0.08	19.82	141.00
0019	25-39	Dark brown (10 YR 4/3); gravelly clay;	50.00	24.24	25.76	7.76	0.07	33.50	8.00	0.46	0.05	22.55	69.00
0019	39-1000	Weathered Granitic	0	0	0	0	0	0	0	0	0	0	0
0021	0-18	Reddish brown (5 YR 4/4); sandy clay	18.60	13.00	68.40	7.12	0.06	5.50	2.50	4.27	1.12	3.72	103.00
0021	18-37	Reddish brown (5 YR 4/4); clay; mode	31.14	18.80	50.06	6.77	0.02	10.00	8.00	4.25	1.09	6.77	100.00
0021	37-78	Dark reddish brown (5 YR 3/4); clay; n	34.60	19.25	46.15	6.72	0.02	10.50	12.00	3.75	1.17	4.67	108.00
0023	0-17	Brown (10 YR 5/3) clay loam; weak, fi	35.00	22.50	42.50	8.00	0.85	14.50	3.26	3.34	0.58	2.80	174.00
0023	17-39	Pale brown (10 YR 6/3) clay loam; we	38.00	19.50	42.50	8.30	0.86	15.81	3.51	3.52	0.62	3.50	158.00
0023	39-42	Yellowish brown (10 YR 5/4) sand; we	4.85	7.65	87.50	6.90	0.10	2.15	0.86	0.45	0.54	4.00	56.00
0023	42-96	Yellowish brown (10 YR 5/4) sand; we	13.70	5.00	81.30	7.00	0.05	4.59	1.53	0.35	0.20	2.00	35.00

Figure 3. The Description table.

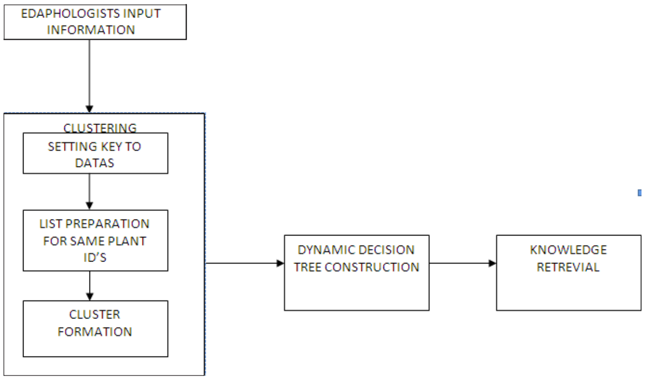

Figure 4. Data flow of the proposed work.

Table 1. Plant dataset

Id	Id of the soil
Name	Name of the soil
Geology	Geological Value
Taxonomy	Taxonomy structure of the plant

Table 2. Soil characteristics description

Id	Id of the soil
Depth	Depth of the Plant
Desc1	Description
Clay	Clay level
Silt	Silt value
Sand	Sand Value
PH	PH value
EC	EC Value
Ca	Calcium Content
Mg	Magnesium Content
Na	sodium Content
K	Potassium

It involves two steps of type conversion and normalization. Here, initially the mean values of the attributes are computed and subsequently type converted and normalized. The description table consists of 17 soil features in which some are in the alphanumerical format. When the input is in the form of alphanumeric value, the system would not be able to access it; hence the value is converted to numeric values. The soil features are transformed to unsigned integer format.

2.1.1 Type Conversion

Categorical variables like soil color, strength and moisture are given in string format of multiple words like "light brownish grey". The categorical value is separated into a single word and every word is allocated a single byte. For every categorical variable, a lookup table is employed to substitute categorical values with numeric values. The conversion of the categorical data to numeric value is carried out as mentioned in the examples.

Example 1: "*Light Brownish Grey*" is coded as 0000 0001 0000 0001 0000 0001=2^16 + 2^8 +2^1
Example 2: "*Light Yellowish Brown*" is coded as 0000 0010 0000 0001 0000 0010=2^16 + 2^9+2^2

Finally, the categorical values are transformed into numeric values in the respective columns and a 17-column vector each with numerical values is obtained.

2.1.2 Normalization

After type conversion, normalization is carried out to confine the values of diverse ranges to a defined specific range. Here, the values from all ranges are converted to a defined range of zero to one. After normalization, 7 column vectors (for 17 features of plant and soil) where each column value ranges between 0 and 1 are obtained. The normalization technique employed in this system is the min-max normalization technique which is used to have the feature values in the range of 0 to 1.

The Pseudo code of the min-max normalization technique is given below:

For i = 1: *Na*
$Min_{old} = Find\ Minimum(i)$
$Max_{old} = Find\ Minimum(i)$

$$Ni = \left({(i - Min_{old})}\big/{(Max_{old} - Min_{old})} \right) \times \left(Max_{new} - Min_{new}\right) + Min_{new}$$

Where, Max_{new} - *specified maximum value to be converted*
Min_{new} - specified minimum value to be converted
Max_{old} - Original max value of the variable
Min_{old} - Original min value of the variable
Ni = Normalized values.

2.2 Clustering and Dynamic Decision Tree Construction

Generally clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Here a list is prepared with the plant IDs and it is stored in separate structures. And for each set a key is set along with their characteristics values.

Plant ID's – P_1, P_2, P_3…… P_n
Ph set- corresponding Ph value + key
Depth set –corresponding depth value + key

The process undergoes pre-pruning of decision tree and hence the size of the decision tree will be very less and can fit into main memory. Further selection of attributes may reduce the size of the tree even better. The lists are formed for the unique plant id's.

Ph value 6.61 is for the plant id's $P_{7,}$ P_{24}, P_{250}
Ph value 7.52 is for the plant id's $P_{2,}$ $P_{28,}$ P_{29}
Ph value 6.52 is for the plant id's $P_{9,}$ $P_{3,}$ P_{240}

The plant ids are inserted into the list. These unique sets are identified using a formula,

Current node * 2 + n
For 0 = 0*2 + 1 = 1
0*2+2 = 2
For 1 = 1*2 + 1 = 3
1*2 + 2 = 4

As shown in Figure 5, Now the sets are (1, 2) and (3, 4). After that the dynamic decision tree is constructed for every set. The tree is stored in the form of binary text file which consists of the plant information and soil characteristics.

2.3 Dynamic Decision Quad tree Construction

Once the dynamic decision quad tree is constructed for storage the quad trees are then constructed as shown in Figure 6. The algorithm for the Dynamic decision Quad tree construction is given below.

Algorithm for Constructing Dynamic decision quad Tree

In the code some representations are made and are defined as:

pta – Point which is to be added into the tree.
noid - The node identification number.

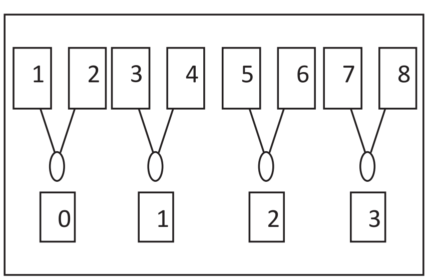

Figure 5. Key values for unique plant set.

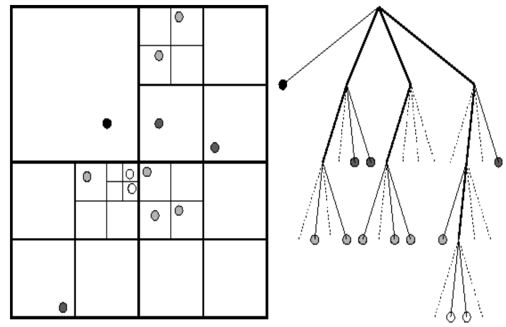

Figure 6. Representation of dynamic decision quad tree.

dis – The distance between node's coordinates and point's coordinates.
th - It represents the threshold value set to add a point to the node.

```
Flag = 0;
 Function add Point (point Pta)
  While (! flag)
    Noid = searchnode(pt, Noid)
    Dis = distance(pt, Noid)
    if(dis > Th)
      Splitnode(Noid)
    Else
      Addpoint node (pt, Noid)
    Flag = 1;
    End If
    End While
```

2.4 Retrieval of Plant Data

From the constructed dynamic decision quad tree the plants names are retrieved and tested. In this paper, effective knowledge retrieval is carried out with the aid of XML architecture and XML tags. The system is designed for retrieving best plants for the input soil condition given by the user. When the user gives a query with input features, the suited plants are retrieved from the XML Quad tree. The information given by our system would benefit Edaphologists in knowing the right plants to be grown for the input soil conditions. That is right plant for the right soil characteristics are obtained with the aid of the system. The pseudo-codes for knowledge retrieval from XML and XML search are given below:

```
Function Search xml_element(p [], mi)
  mx[4,2], Element Ele[4];
  Element.El = XML.GetElementByID(mi)
  While(El.Getattribute(ital ISPARENT) = 1) nodule
```

Ele[1] = E11.NextChildElement(ital TopLeft)
Ele[2] = E12.NextChildElement(ital TopRight)
Ele[3] = E13.NextChildElement(ital BottomLeft)
Ele[4] = E14.NextChildElement(ital BottomRight)
$mx[1,:] = El1.GetAttributes(A,B)$
$mx[2,:] = El2.GetAttributes(A,B)$
$mx[3,:] = El3.GetAttributes(A,B)$
$mx[4,:] = El4.GetAttributes(A,B)$
$id = findnearest_{nodule_{id}}(pt, mx)$
$mi = Ele[id].GetAttribute(ital\ NID)$
$El = Xml.GetElementByID(mi)$
End While
Return mi
End function

3. Results and Discussions

The sample input query and its corresponding output is shown in Table 3. For analyzing the performance of the implemented system, six input queries are used as shown Table 4 and their outputs are analyzed. Table 5 shows the number of plants retrieved, computation time and memory usage for the sample input queries. Figure 7 shows the number of plants retrieved for our proposed method and for our previous methods. Figure 8 shows the computation time for our proposed method and previous methods and Figure 9 shows the memory usage by our

Table 3. Sample input query and its output

Input Query	*Output*
Depth: 0–18 *Color: red* *Sand: sandy clay loam* *Strength: moderate medium sub angular blocky* *Moist: slightly sticky* *Pores: common pores* *Clay: 24.60* *Silt: 15.2* *Sand: 60.2* *PH: 7.36* *EC: 0.04* *CA:8.0* *Mg: 1.5* *Na: 0.68* *K: 0.32* *P2O5: 29.45* *K2O: 228.00*	*Name: Palmyrah* *Geology: Granite* *Taxonomy: Fine, mixed, isohyperthermic, noncalcareous, Typic Rhodustalfs*

Table 4. Input queries

Query1	Query2	Query3
Depth =26-52 Color = light grey to grey Sand = sandy clay Strength = medium moderate sub angular blocky Moist = sticky Pores = pores Clay =36.31 Silt =13.51 Sand=50.18 PH=6.28 EC=9.40 CA=9.40 Mg=1.80 Na=0.50 K=0.12 P2O5=11.26 K2O=108.00	Depth =50-91 Color = brownish yellow Sand = clay Strength = medium moderate sub angular blocky Moist = sticky Pores = few pores Clay =38.89 Silt =20.25 Sand=50.18 PH=8.65 EC=0.09 CA=15.34 Mg=7.82 Na=3.01 K=0.12 P2O5=14.00 K2O=302.00	Depth =13-33 Color = reddish brown Sand = sandy clay Strength = medium weak sub angular blocky Moist = sticky Pores = few pores Clay =47.00 Silt =22.00 Sand=8.00 PH=8.00 EC=1.50 CA=14.00 Mg=10.00 Na=1.10 K=0.60 P2O5=7.00 K2O=208.00
Query4	**Query5**	**Query6**
Depth =41-51 Color =very pale brown Sand =sandy Strength =medium sub angular blocky Moist = Pores = Clay =3.40 Silt =4.00 Sand=92.60 PH=8.00 EC=0.06 CA=3.57 Mg=0.51 Na=0.25 K=0.10 P2O5=4.00 K2O=49.00	Depth =23-37 Color = dark red Sand =sandy clay Strength =moderate medium sub angular blocky Moist =slightly sticky Pores =few fine pores Clay =40.00 Silt =24.00 Sand=36.00 PH=7.17 EC=0.16 CA=10.00 Mg=4.50 Na=1.28 K=0.93 P2O5=25.65 K2O=195.00	Depth = Color =dark red Sand =sandy clay Strength =moderate medium sub angular blocky Moist =slightly sticky Pores =few fine pores Clay =40.00 Silt =24.00 Sand=36.00 PH=7.17 EC=0.16 CA=10.00 Mg=4.50 Na=1.28 K=0.93 P2O5=25.65 K2O=195.00

Table 5. Plants retrieved and computation time for different queries

Performance Metrics	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6
Plants Retrieved	32	31	27	42	34	33
Computation time(ms)	102	103	102	105	98	104
Memory usage(kb)	2486	2487	2487	2487	2485	2486

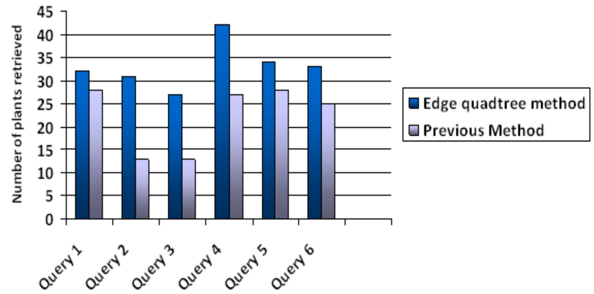

Figure 7. Chart showing the number of plants retrieved for various queries by the two methods.

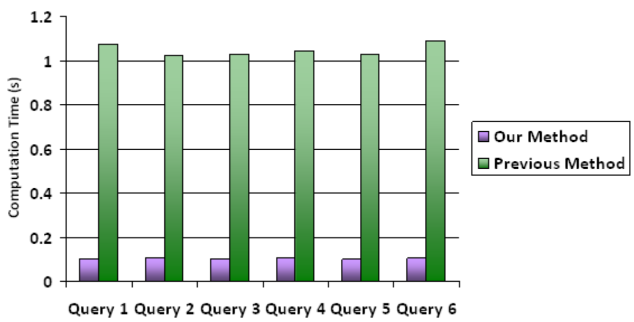

Figure 8 Chart showing the computation time for various queries by the two methods.

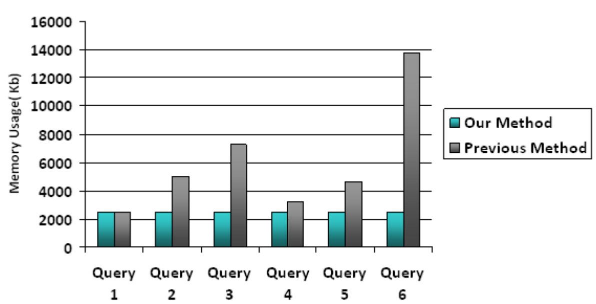

Figure 9. Chart showing the memory usage for various queries by the two methods.

proposed method and previous methods. It can be seen that an average of 33 plants are retrieved per user query. Maximum value came about 42 for 4th query. The system consumed an average of 2486 Kb per query. Average computation time came about 102 ms. High number of plants retrieved, low memory usage and low computation time further demonstrates the effectiveness of the system.

4. Conclusion and Future Work

Clustering and Dynamic decision tree technique is used in this work for storing data in edaphology. The proposed system proves better storage when compared to the previously existing systems. Here the storage is in the form of binary text file which reduces the memory space and allow more number of plants to be stored. This would help the Edaphologists and agriculturists in obtaining more appropriate results at the time of retrieval. Here, dynamic decision quad trees are used and better retrieval is achieved for various inputs.

The future scope of our work is to employ still better technique for storage and retrieval for getting relevant plant names. The new technique may be designed with the aim of retrieving more number of plants for the input soil characteristics and it should be with low memory usage and low computation time.

5. References

1. Ras E, Rech J. Using Wikis to support the Net in improving knowledge acquisition in capstone projects. International Journal of Systems and Software. 2009; 82(4):553–62.
2. Wang YJ, Xin Q, Coenen F. Hybrid Rule Ordering in Classification Association Rule Mining. International Journal of Transactions on Machine Learning and Knowledge Mining. 2008; 1(1); 1–15.
3. Egecioglu O, Ferhatosmanoglu H, Ogras U. Dimensionality reduction and similarity Computation by inner-product approximations. IEEE Transactions on Knowledge and Data Engineering. 2004; 16(6):714–26.
4. Ozbayrak M, Bell R. A knowledge-based decision support system for the management of parts and tools in FMS'. International Journal of Decision Support Systems. 2003; 35(4):487–15.
5. Prasad R, Bihar ABR, Sinha AK. KISAN: An Expert System for Soil Nutrient Management. AFITA 2002: Asian agricultural information technology and management. Proceedings of the Third Asian Conference for Information Technology in Agriculture; Beijing: China. 2002. p. 26–28.
6. Denning S. The role of ICT's in knowledge management for development. The Courier ACP-EU; 2002. p. 58–61.
7. Wang Y, Wang J, Zhang S. Collaborative knowledge management by integrating knowledge modeling and workflow modeling. Proceedings of IEEE International Conference on Information Reuse and Integration; IRI'05; 2005. p. 13–18.
8. Chen SK. An Exact Closed-Form Formula for D-Dimensional Quadtree Decomposition of Arbitrary Hyperrectangles. IEEE Transactions on Knowledge and Data Engineering. 2006; 18(6): 784–98.
9. Sikder IU. Knowledge-based spatial decision support systems: An assessment of environmental adaptability of crops. Expert Systems with Applications: An International Journal. 2009; 36(3):5341–47.

10. Abel M, Silva LAL, De Ros LF, Mastella LS, Campbell JA, Novello T. Petrographer: Managing peterographic data and knowledge using an intelligent database application. Expert systems with Application. 2004; 26(1): 9–18.
11. Bui EN, Henderson BL, Viergever K. Knowledge discovery from models of soil properties. Ecological- Modelling. 2006; 191(3-4): 431–46.
12. Faloutsos C, Jagadish HV, Manolopoulos Y. Analysis of the N-Dimensional Quadtree Decomposition For Arbitrary Hyperrectangles. IEEE Transactions on Knowledge and Data Engineering. 1997; 9(3):373–83.
13. Venkatesan E, Velmurugan T. Performance Analysis of Decision Tree Algorithms for Breast Cancer Classification. Indian Journal of Science and Technology. 2005; 8(29):1–8.