Clustering Web Pages Based On Doc Type Structure In a Distributed Manner

BRAC
UNIVERSITY

Inspiring Excellence

Conducted By:
Kazi Samiul Kader - ID: 11101011
Sagufa Nawar - ID: 11101007
Nusrat Sharmin Ananna - ID: 11101031
Sarah Khan - ID: 11201022

Supervisor:
Moin Mostakim

Declaration

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor

Signature of Author

Moin Mostakim

Kazi Samiul Kader

Sagufa Nawar

Nusrat Sharmin Ananna

Sarah Khan

Acknowledgement

This thesis was suggested by Mr. Moin Mostakim, SECS Department, BRAC University. This thesis is the work of Kazi Samiul Kader, Sagufa Nawar, Nusrat Sharmin Ananna and Sarah Khan, students of SECS Department of BRAC University. This thesis paper data has been collected from
https://www.youtube.com,
http://dailymotion.com,
http://skysports.com/football,
http://www.espnfc.com,
http://stackoverflow.com.
All the data has been collected by our own implemented crawler.
This thesis also takes help of the Weka software from
http://www.cs.waikato.ac.nz/ml/weka.
We wish to acknowledge the efforts of our supervisor Mr. Moin Mostakim for his contribution, guidance and support in conducting the research and preparation of the report. We also wish to acknowledge our chairperson, Professor Md. Haider Ali for his direct or indirect support and inspiration. Finally, we are very thankful to BRAC University, Bangladesh for giving us a chance to complete our B.Sc degree in Computer Science and Engineering

Contents

1 Abstract

Web page clustering is an important part of modern web technology. By structuring similar web pages together we can find related information, suggest similar choices etc. All modern search engines depend on web page clustering. It is interesting to work on this topic as it presents a novel academic challenge and also practical application. In this thesis we clustered web pages by using the HTML tag structure of web pages. We represented each web page as a vector of tag percentages and clustered them using k-means clustering algorithm and DBSCAN clustering algorithm. We selected k-means and DBSCAN algorithm because they are well known clustering algorithms and also they have not been applied together and compared in the field of web page clustering as we did in this thesis. After clustering on three different category of five websites in three stages, both algorithms produced over minimum 88% accuracy in clustering compared to the original clusters. In this process we used the weka data mining software, because it is well tested in terms of accuracy and efficiency. It is also open source.

2 Introduction

World Wide Web has become the part of our daily life. It is also the source of unimaginable data and information. Finding patterns in data is as important as the data itself. But we cannot attempt to find patterns in this whole World Wide Web all at once. For achieving some degree of success we need to consider sections of the whole Web. But how do we section the whole web?

Here is where web clustering comes in. We section the whole web by partitioning it by basis of similar sections. To find those similar sections we need to cluster similar web pages together. The formed clusters will represent a similar section of the web.

But web page clustering presents a problem itself. There is no deterministic way to say that this web page is similar to that web page. Even a single web site with many pages has different pages in it. So how does we make sure that our clustered web page has reasonable amount of accuracy? Are we actually looking at the similar pages?

We tried to answer this question by taking the web page HTML tag structure, presenting it as a multidimensional vector and giving that vector data set to two well known clustering algorithms - k-means and DBSCAN. Then we compared the outcome of those algorithms and compared to our known clusters to find out if they can produce satisfactory amount of accuracy.

2.1 Motivation

Today our whole world is dependent on World Wide Web. We are connected to it. In recent years web based technology has seen the rise of successful companies such as Google, Facebook etc. This success drives new invention in the area of web based technology. So we are also very motivated to work on such a promising field.
Besides that, web clustering is fascinating in its own right. As the solution is not deterministic, there are whole new ways where machine learning techniques and algorithms can be applied. We also had the chance to experiment with different web page representation and applying our collected data set to two well known clustering algorithms. The outcome of such experiments is intellectually motivating.

2.2 Overview of Thesis

We have discovered some steps to successfully perform web clustering. The major steps are-

1. To collect web pages from five websites which fall in three categories

2. Extract feature vector of HTML tag percentages

3. Cluster our data set using k-means algorithm with weka software

4. Cluster our data set using DBSCAN algorithm with weka software

All of these steps require extensive literature review, experimentation and research to produce acceptable end result. So the next sections will describe each of these steps in more detail.

3 Distributed Crawler

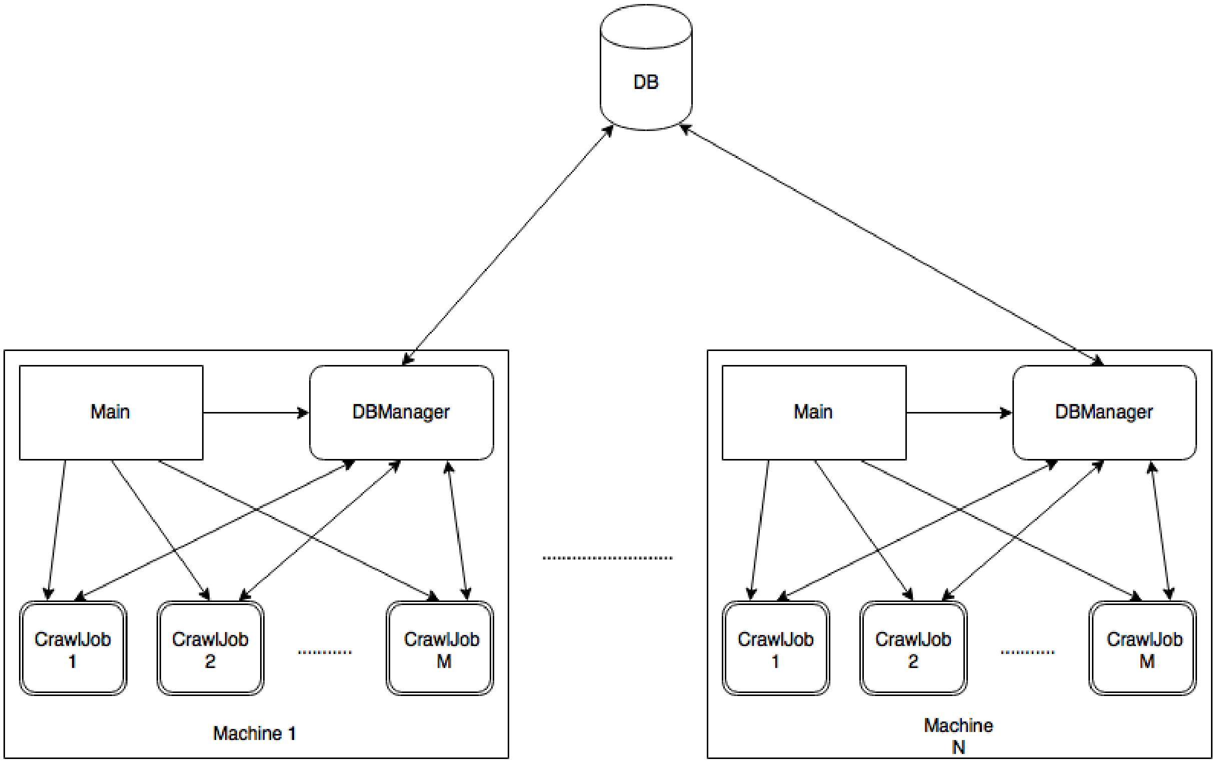
For collecting the web pages we have implemented our own distributed crawler. The motivation behind implementing our own crawler instead of using some well established crawler is to have the flexibility to add our feature extraction procedure and also to make it suitable to our experiment environment.

3.1 Literature Review

Before attempting to make a distributed crawler which suits our own need. We looked at previous works on this subject. We looked at distributed crawler which can work on networked workstations and fetch several hundred pages per second [1]. We also looked at focused crawler which can find selective web pages based on a topic using exemplary documents [2]. As Google is now the pioneer of web technologies, we looked at the first Google crawler [3]. Finally we looked at scalable web crawler which can scale up to the whole web [4]. These literature reviews prepared us to tackle our goal of building a distributed web crawler of our own.

3.2 Overall Crawler Architecture

To describe the overall architecture of the crawler, we must describe the major components of the crawler and describe how these components are connected to each other.



3.2.1 The Database

We used a MySql database as our central database. It resides on one machine which other machine database managers can refer to by IP address. If one of the database manager is in the same machine, it refers to the database manager by localhost. The database has two tables- visited and unvisited. Each table has one column to store the URLs of the visited and unvisited web pages.

3.2.2 The Main Program

The main program is the starting point of the crawler. It is responsible for starting the database manager. It also starts M number of CrawlJobs. The number M is provided as the argument to it when starting the main program. The main program is also responsible for starting a CrawlJob whenever one of the M CrawlJob finishes and dies. So at a particular moment in time, there is always M CrawJobs running. And one second delay is given after one CrawlJob is started after the completion of any previous CrawlJob, we consider this 1 second as politeness delay.

3.2.3 The Database Manager

When the main program starts the database manager, it loads a configuration file where the address of the database is given with username and password to access the database. The database manager holds function for reading and writing to the database. These functions are-

1. Inserting in the visited and unvisited tables

2. Deleting from the visited and unvisited tables
3. Querying if a URL is present in visited or unvisited tables

3.2.4 CrawlJobs

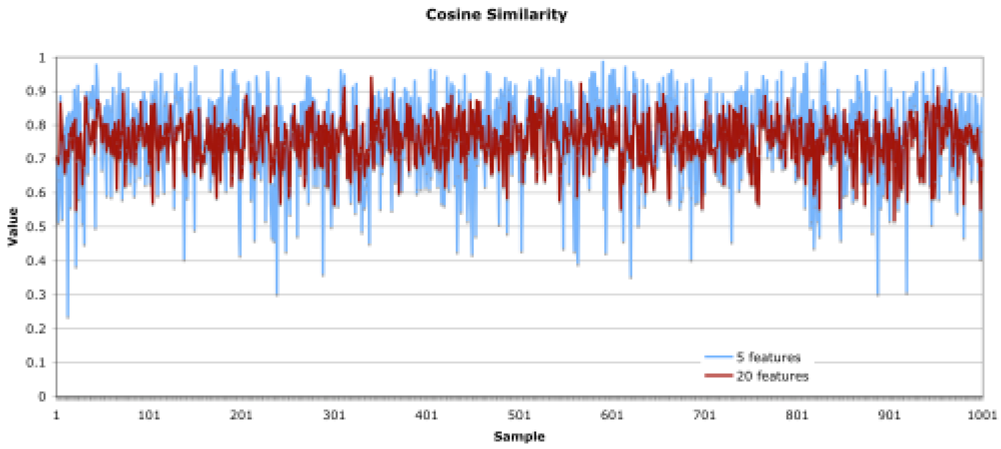
CrawlJobs are the main work horse of the crawler. They are Java thread which is started by the main program. When the main program starts a CrawlJob, it passes a reference of the database manager to that CrawlJob. At first, the CrawlJob requests the database manager to fetch one URL from the unvisited table of the database. Then when it receives the unvisited URL, it downloads the web page and extracts all the outgoing link. Then it gives all the outgoing links to the database manager to be saved in the unvisited table for further processing by other future CrawlJobs. It also extracts the feature vector and saves it in .arff format which is needed for clustering with weka software. Finally it gives the database manager confirmation that the current URL has been finished processing and the database manager can save this URL in the visited table so no other future CrawlJob can re-crawl this URL. Then the main program finishes this CrawlJob and starts another CrawlJob instead of its place.

4 Features

The first stage of clustering is data preparation. It is important to prepare data set and then having it to transform to few important number of attributes which will become more useful and helpful to learning algorithm. When the data set, in this case the web page, have such a large amount of features, which are the tags, it is very important and beneficial to have it reduced to certain number number of features so that the learning algorithm can work more efficiency on it. We are using feature vector as a representation of web page [5]. The tags in the web page are the features that we are using as a set of data for the analyzing and clustering. 20 HTML element tags are used for this thesis and those tags are represented as 20 dimensional vector. Those 20 dimensional tags has been reduced to 10 dimensional vector through a process of feature extraction and feature selection which are part of dimension reduction techniques. Before going into how dimension reduction is done, let's look at the reason why dimension reduction is done. Historically, before dimension reduction was ever developed, the traditional learning algorithm has to run all over the whole data set and this led to degradation in the performance of a given learning algorithm as the number of features increases and not to mention time consuming since it has to scan the whole data set for the required features. This was referred as, by Bellman, ”curse of dimensionality” [7].

Now as to why dimension reduction is done, the benefits are as follow:

1. Tasks such as classification or clustering can often yield more accurate and readily interpretable results [5].
2. The identification of a reduced set of features that are predictive of outcomes can be very useful from a knowledge discovery perspective [5].
3. Reduced Training time: Less data means that algorithms train faster [6].
4. Reduces Over fitting: Less redundant data means less opportunity to make decisions based on noise [6].
5. Noisy or irrelevant features can have the same influence on classification as predictive features so they will impact negatively on accuracy [5].
6. Look more similar on average the more features used to describe them. Resolution of a similarity measure can be worse in 20D than in a 5D space [5].



The more dimensions used to describe objects the more similar on average things appear. This figure shows the cosine similarity between randomly generated data objects described by 5 and by 20 features. It is clear that in 20 dimensions similarity has a lower variance than in 5.

Now let get back to the process of how those tags are extracted. There are two process of doing that

1. Feature Selection
2. Feature Extraction

Feature Selection methods choose features from the original set based on some criteria, Information Gain, Correlation and Mutual Information are just criteria that are used to filter out unimportant or redundant features. Embedded or wrapper methods, as they are called, can use specialized classifiers to achieve feature selection and classify the data set at the same time.

Feature Extraction methods are transformative, that is you are applying a transformation to your data to project it into a new feature space with lower dimension. PCA, and SVD are examples of this.

4.1 Feature Extraction

We are going to extract tags from three different kind of web pages.

1. Forum
2. Video
3. Sport

Feature extractor receives a web page from the crawler, then it extracts the specified feature and discards the page. As for the feature, we decided that the structure of each web page can be used as the clustering feature. The feature extractor is designed to take the ratio twenty HTML element tags in each page. The tags are-
Input, option, h1, h2, h3, h4, div, a, p, span, script, noscript, img, section, form, link, ul, ol, class. That means we are representing each page as a vector of twenty elements. The formula for finding the ratio is - number of a particular tag occurrence / total no of tags. For example- a web page can be represented as (0.3, 0.1, 0.05, 0.2, 0.9, 0.5, 0.6). One thing to note here

is that as we are taking the ratio of each element the ratio will always be within range 0 to 1 inclusive. And the sum of all vector elements is always less than or equal to 1.

4.2 Feature Selection

Now that all the tags has been selected which in this are 20 tags, now we are going to separate the tags into important tags and non important tags. We are using weka for this purpose.
According to weka website, ”Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a data set or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes” [6].
We are using weka because-

- It is open source and freely available
- It is platform-independent
- It is easily usable by people who are not data mining specialists.
- It provides flexible facilities for scripting experiments

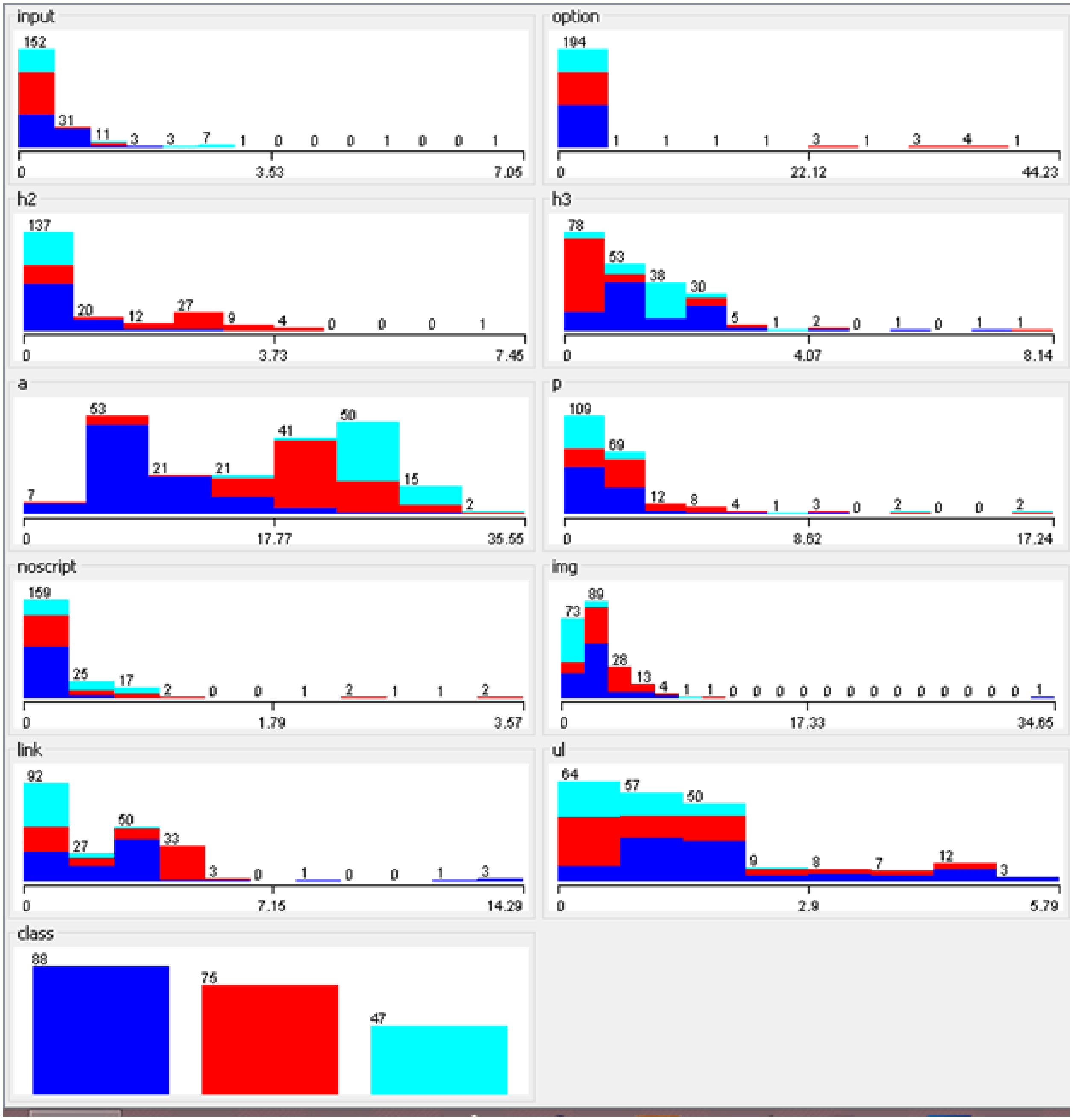
- It provides many different algorithms for data mining and machine learning

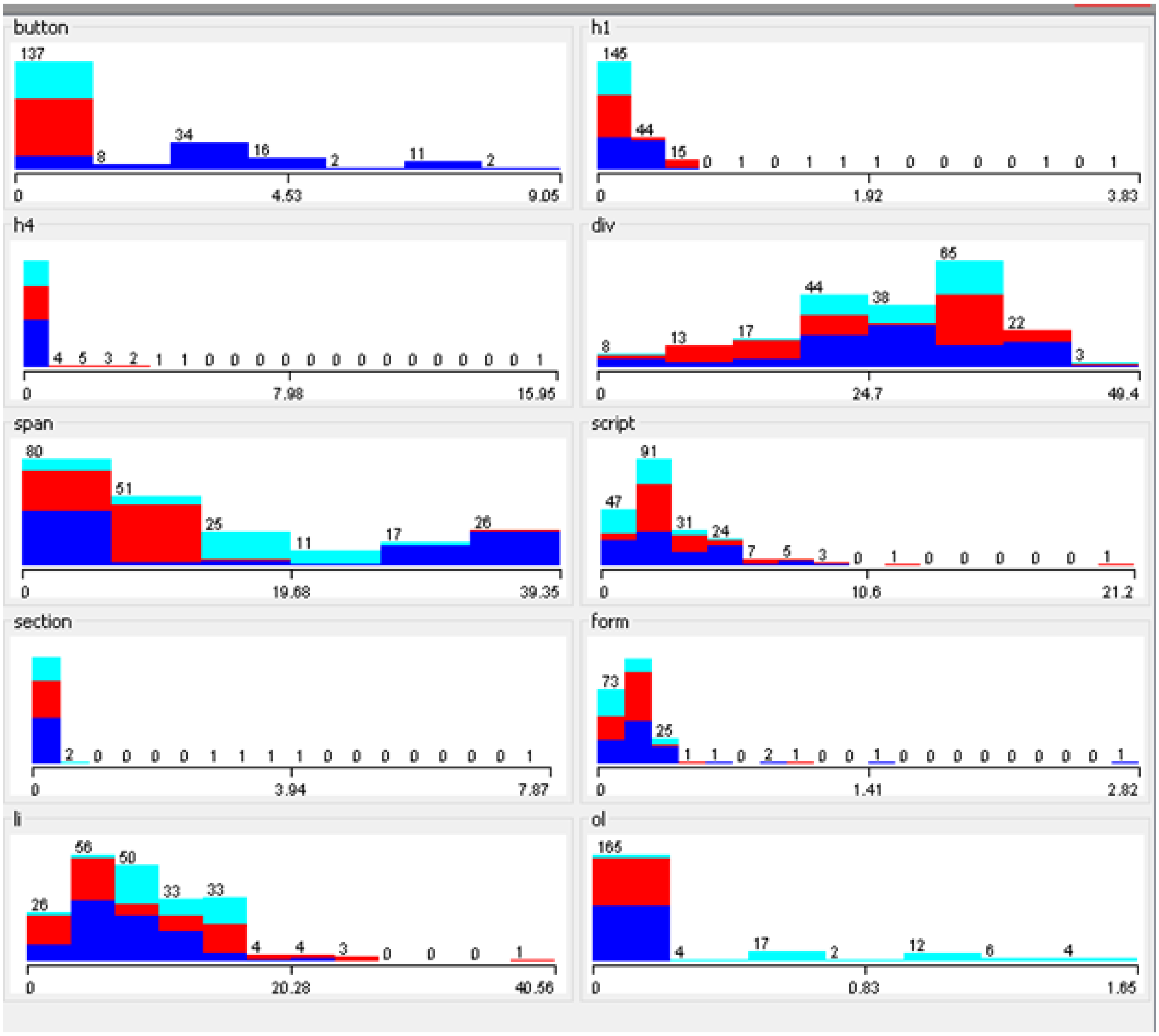
- It has kept up-to-date, with new algorithms being added as they appear in the research literature.

The tags extracted were loaded to the weka software and the output was as display on the next page.

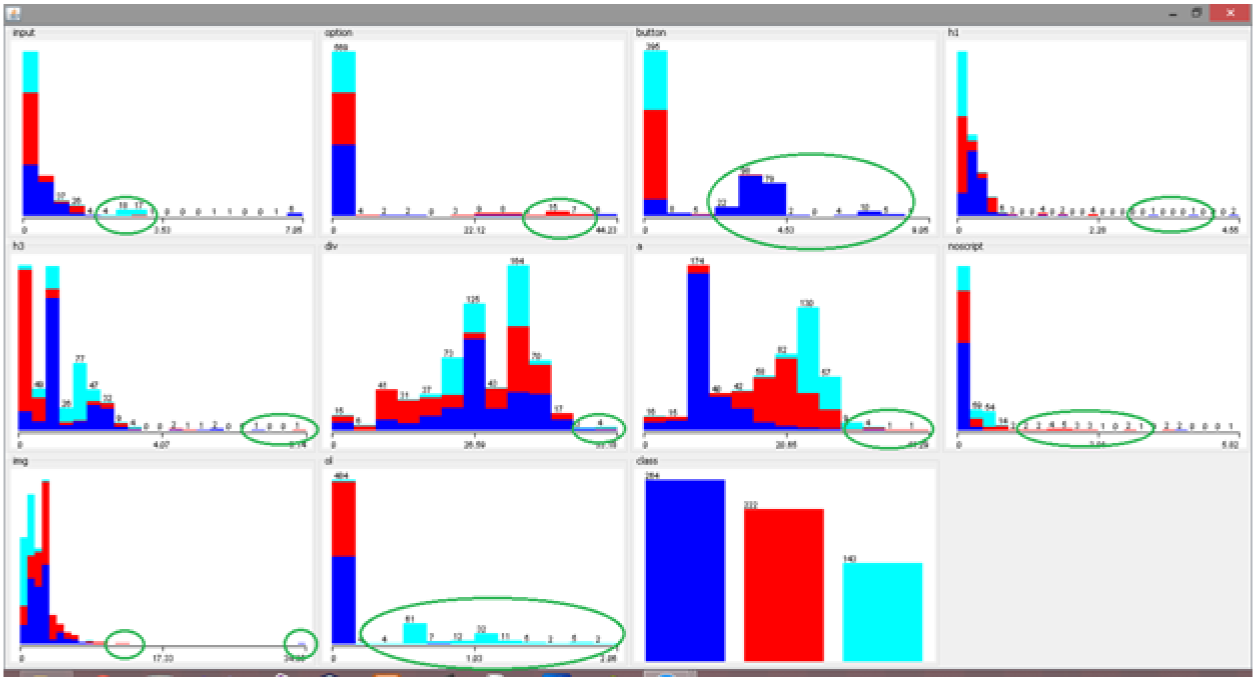
These are the tags extracted from the three web page. Color indicate the different type of web page that tags was extracted from. X axis represents percentage from 1 to 100 and y axis is number of data point.

We are going to separate the tags to good and bad tags through visual inspection. If we look at the graph, the graph with more variation between data points(the data which are more scattered) are under good tags. And the one with less distinction of data point(that is with close data point) are under bad tags.


input
152
31
11
3
3
7
1
0
0
0
1
0
0
1
0
3.53
7.05
option
194
1
1
1
1
3
1
3
4
1
0
22.12
44.23
h2
137
20
12
27
9
4
0
0
0
1
0
3.73
7.45
h3
78
53
38
30
5
1
2
0
1
0
1
1
0
4.07
8.14
a
7
53
21
21
41
50
15
2
0
17.77
35.55
p
109
69
12
8
4
1
3
0
2
0
0
2
0
8.62
17.24
noscript
159
25
17
2
0
0
1
2
1
1
2
0
1.79
3.57
img
73
89
28
13
4
1
1
0
0
0
0
0
0
0
0
0
0
0
0
0
0
1
0
17.33
34.65
link
92
27
50
33
3
0
1
0
0
1
3
0
7.15
14.29
ul
64
57
50
9
8
7
12
3
0
2.9
5.79
class
88
75
47



For example In first graph which is the tag $< li >$, we can see red and blue data point are closely mixed together and there are not much data points that are distinctly separated from one another. Therefore, it is under bad tag.



These are final 10 tags which are under good tags. The green circle drawn mean attributes which are distinctly separated.

The good tags are going to be used for clustering process. And also the 20 dimensional vector are reduced to 10 dimensional vector which is half of the initial vector. And in this ways the cluster engine will work more efficient, faster and accurate.

5 K-Means Clustering Algorithm

In statistic and data mining, k-means clustering is well known for its efficiency in clustering large data sets and grouping them. The aim is to group data points into clusters such that item with similar characteristics are grouped together in the same cluster. In general, given a set of objects together with their attributes, the goal is to divide the objects into k clusters such that objects lying in one cluster should be as close as possible to each other (homogeneity) and objects lying in different clusters are further apart from each other.

5.1 K-Means Algorithm

Let $X = \{x_i, i = 1, ..., n\}$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k, 1, ..., k\}$. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of cluster c_k. The squared error between k and the points in cluster c_k is defined as-

$$J(c_k) = \sum_{x_i \in c_k} \| x_i - \mu_k \|^2$$

The goal of K-means is to minimize the sum of the squared error over all the K clusters.

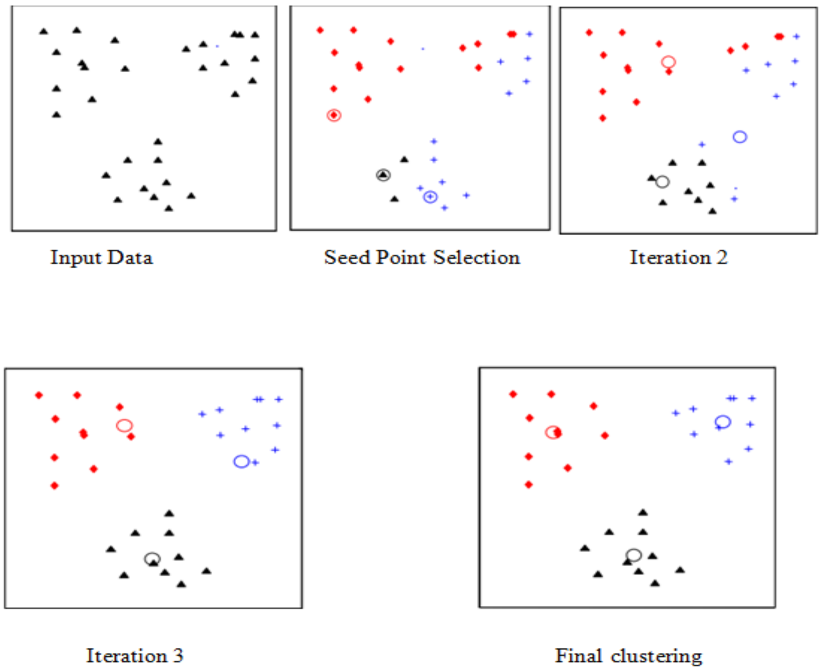
$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \| x_i - \mu_k \|^2$$

K-means is a greedy algorithm which is normally converges to a local minimum. K-means starts with an initial partition with K clusters and assign patterns to clusters in a way that it can reduce the squared error. Since the squared error tends to decrease with an increase in the number of clusters K (with J(C) = 0 when K = n), it can be minimized only for a fixed number of clusters [8], The main steps of K-means algorithm are as follows-

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.

2. Generate a new partition by assigning each pattern to its closest cluster center.

3. Compute new cluster centers.



Input Data

Seed Point Selection

Iteration 2

Iteration 3

Final clustering

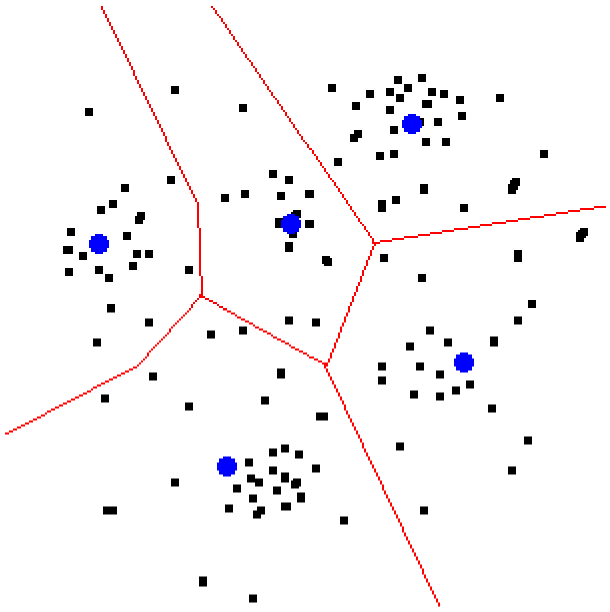
5.2 Parameters Of K-Means

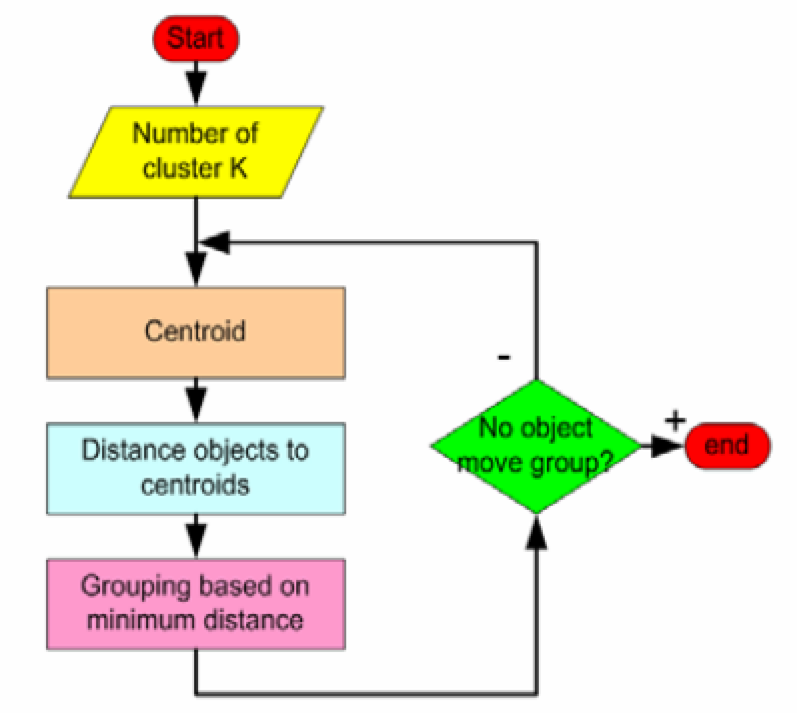
Three user-specified parameters are need for K means algorithm:

1. Number of clusters K
2. Cluster initialization
3. Distance metric

The most important and critical choice is K. While no mathematical criterion exists, a number of heuristics are available for choosing K. Typically, K-means is run independently for different values of K and the partition that appears the most meaningful to the domain expert is selected. Different initializations can lead to different final clustering because K-means only converges to local minima. One way to overcome the local minima is to run the K-means algorithm, for a given K, with several different initial partitions and choose the partition with the smallest value of the squared error. K-means is typically used with the Euclidean metric for computing the distance between points and cluster centers also known as centroids.

A centroid is ”the center of mass of a geometric object of uniform density”.




Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

5.3 Distance Calculation

In K means algorithm Euclidean distance [8] is used normally. Euclidean Distance is the most commonly used distance algorithm. Euclidean distance or simply 'distance' examines the root of square differences between coordinates of a pair of objects.

5.3.1 Formula

$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

5.4 Partitioning Methodologies

- **Dynamically Chosen:** This method is good when the amount of data is expected to grow. The initial cluster means can simply be the first few items of data from the set. For instance, if the data will be grouped into 3 clusters, then the initial cluster means will be the first 3 items of data.

- **Randomly Chosen:** The initial cluster means are randomly chosen values within the same range as the highest and lowest of the data values. It is done randomly.

- **Choosing from Upper and Lower Bounds:** Depending on the types of data in the set, the highest and lowest (or at least the extremities) of the data range are chosen as the initial cluster means.

5.5 Experimentation

We considered the some attributes of web pages as feature vector. As web page contains many tags or attributes. We chose most ten suitable tags to classify our data in K-means cluster. These 10 tags compose ten feature vectors which are used to cluster the web pages in K-means. For clustering we used weka software for Simple K-means clustering. The three different Web page clusters are-

1. Video
2. Sport
3. Forum

Our selected attributes are: input, button, option, div, h1, h3, noscript, a, img and ol. We have tested k-means clustering algorithm over three different sets of inputs of varying size.

- SMALL (210 pages : less dense)
- MEDIUM (628 pages : medium dense)
- LARGE (3939 pages : largely dense)

We followed the steps below for clustering:

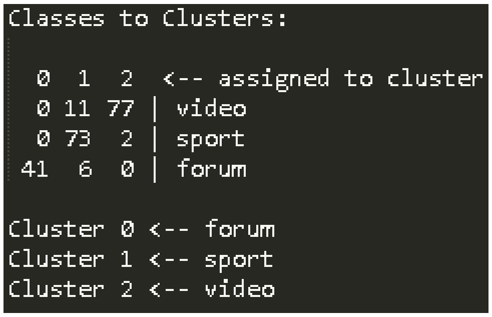
1. First we selected the feature file as input which contains the feature vectors for different attributes
2. Then we chose the number of cluster which is 3 in our case
3. Then we chose a seed value for initial partitioning

Seed value selection is very sensitive for the result of simple K-means algorithm. This seed value determines initial partition and random seed values were given as inputs and results were evaluated [9]. In weka initial partition is chosen via random approach based on seed value. Then the computation is done over three sets feature vectors.

5.5.1 Small Set of Web pages (210)

For small set of data we used five tags button, a, noscript, img and ol. We used 10 as seed value and number of iteration was 9.

Result: K-means clustering generated following result. 0 cluster is assigned for forum, 1 cluster is for sport and cluster 2 is for video. We observe number of misclassified vectors from this result.

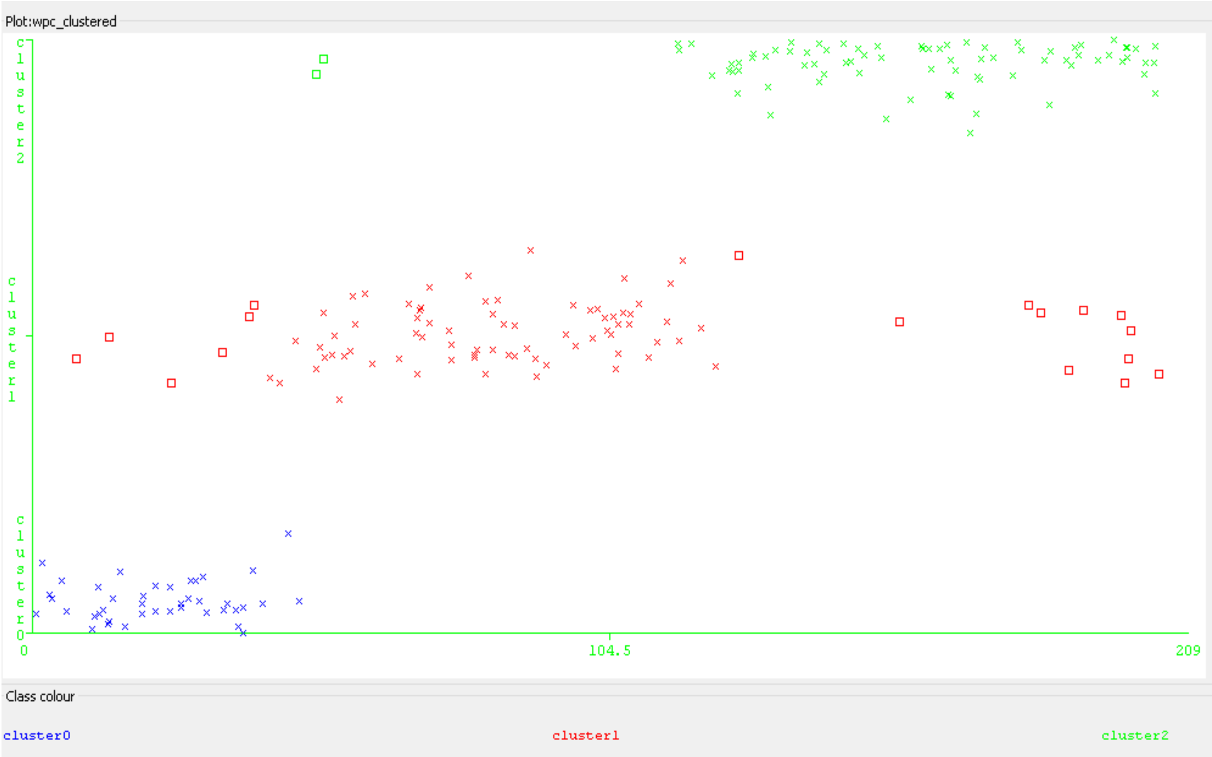


```
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 11 77 | video
  0 73  2 | sport
 41  6  0 | forum

Cluster 0 <-- forum
Cluster 1 <-- sport
Cluster 2 <-- video
```

Cluster	Name	Misclassified
0	Forum	0
1	Sport	17
2	Video	2

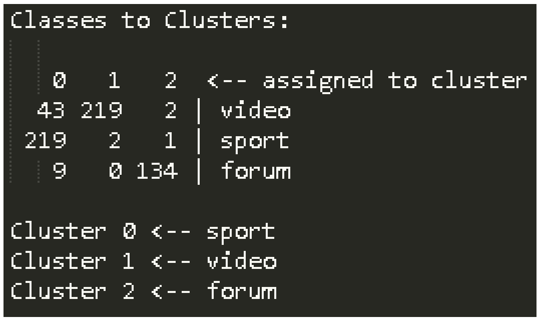


The final accuracy rate for Small Set of Web Page is 90.9524%.

5.5.2 Medium Set of Web pages (628)

For medium set of data we used five tags button, a, noscript, img and ol. We used 10 as seed value and number of iteration was 17.

Result: K-means clustering generated following result. 0 cluster is assigned for sport, 1 cluster is for video and cluster 2 is for forum. We observe number of misclassified vectors from this result.

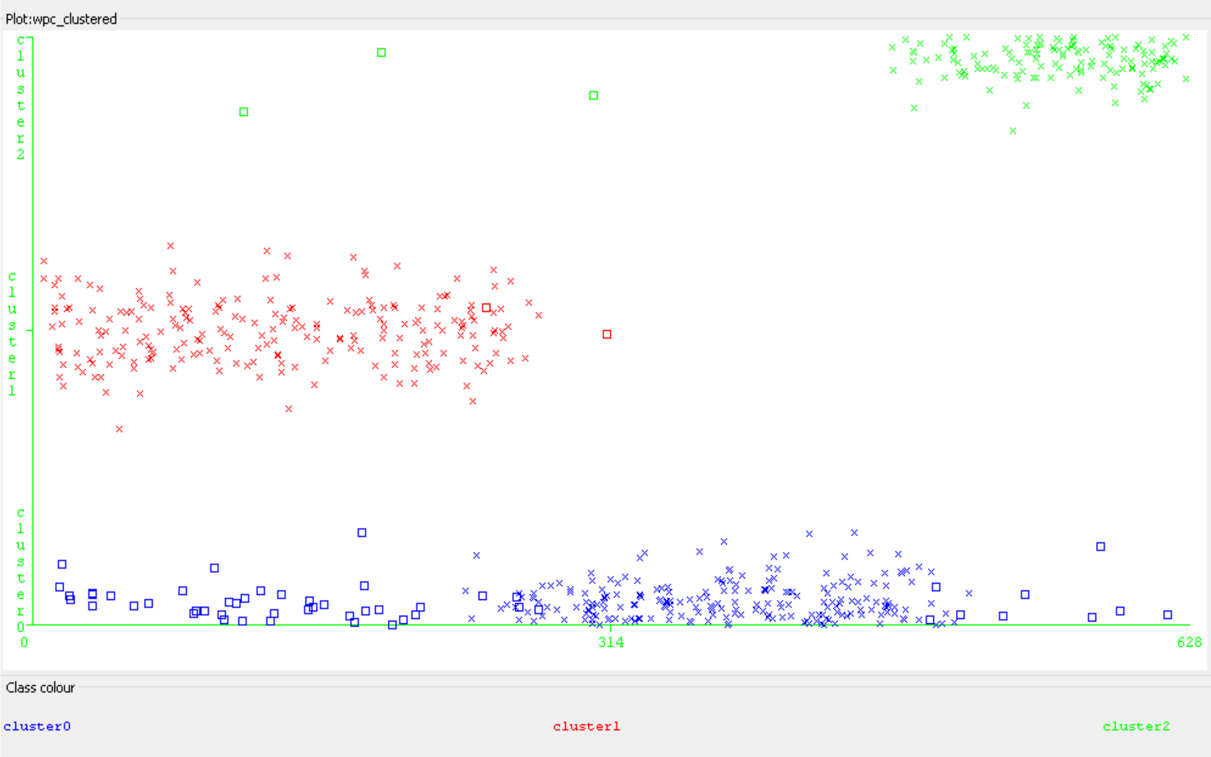


```
Classes to Clusters:

   0   1   2  <-- assigned to cluster
  43 219   2 | video
 219   2   1 | sport
   9   0 134 | forum

Cluster 0 <-- sport
Cluster 1 <-- video
Cluster 2 <-- forum
```

Cluster	Name	Misclassified
0	Sport	52
1	Video	2
2	Forum	3

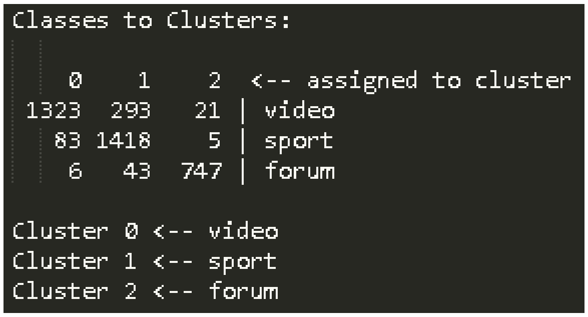


The final accuracy rate for Medium Set of Web Page is 90.938%.

5.5.3 Large Set of Web pages (3939)

For medium set of data we used five tags button, a, noscript, img and ol. We used 20 as seed value and number of iteration was 3.

Result: K-means clustering generated following result. 0 cluster is assigned for video, 1 cluster is for sport and cluster 2 is for forum. We observe number of misclassified vectors from this result.

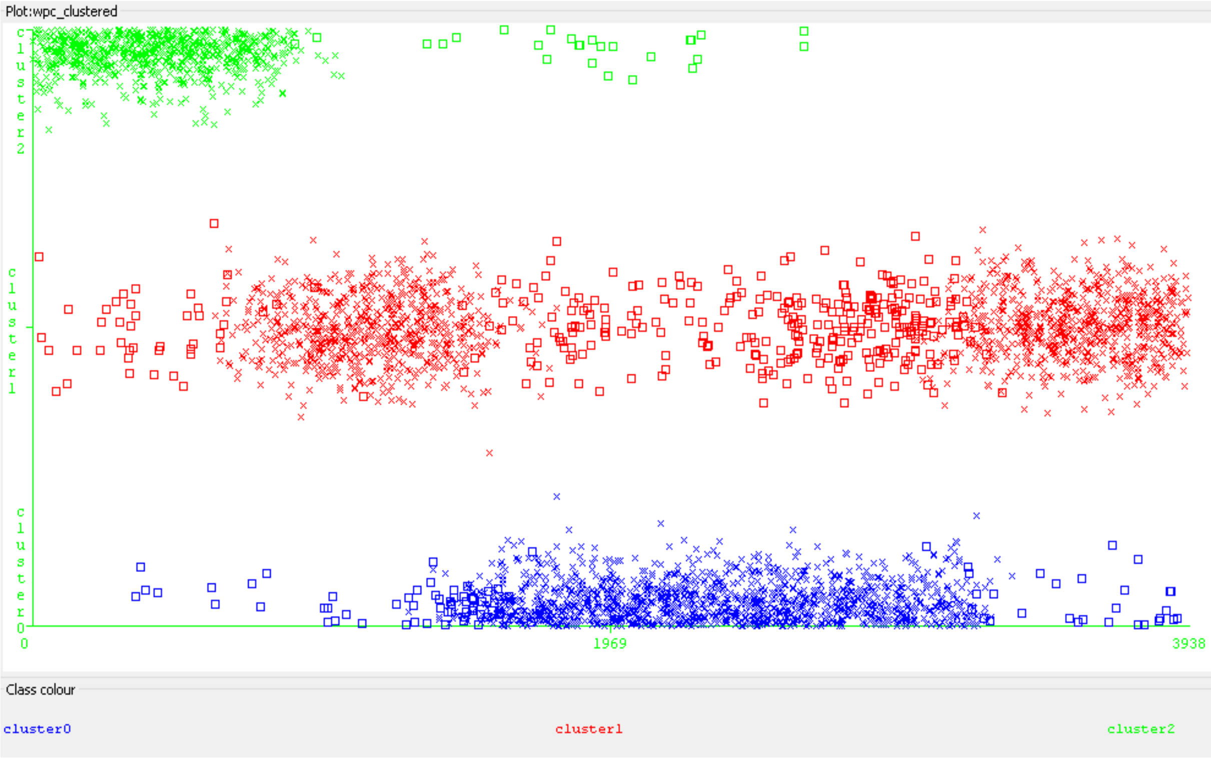


```
Classes to Clusters:

    0    1    2  <-- assigned to cluster
 1323  293   21 | video
   83 1418    5 | sport
    6   43  747 | forum

Cluster 0 <-- video
Cluster 1 <-- sport
Cluster 2 <-- forum
```

Cluster	Name	Misclassified
0	Video	89
1	Sport	336
2	Forum	26



The final accuracy rate for Large Set of Web Page is 88.5504%.

5.6 K-Means Clustering Weaknesses

- With fewer samples of data, initial grouping will determine the cluster significantly.
- The number of clusters, k, must be determined before hand.
- With fewer samples of data, inaccurate clustering can occur.
- We never know which variable contributes more to the clustering process since we assume that each has the same weight.
- The accuracy of mathematical averaging weakens because of outliers, which may pull the centroid away from its true position.[10]
- The results are clusters with circular or spherical shapes because of the use of distance.

5.6.1 Possible Solutions

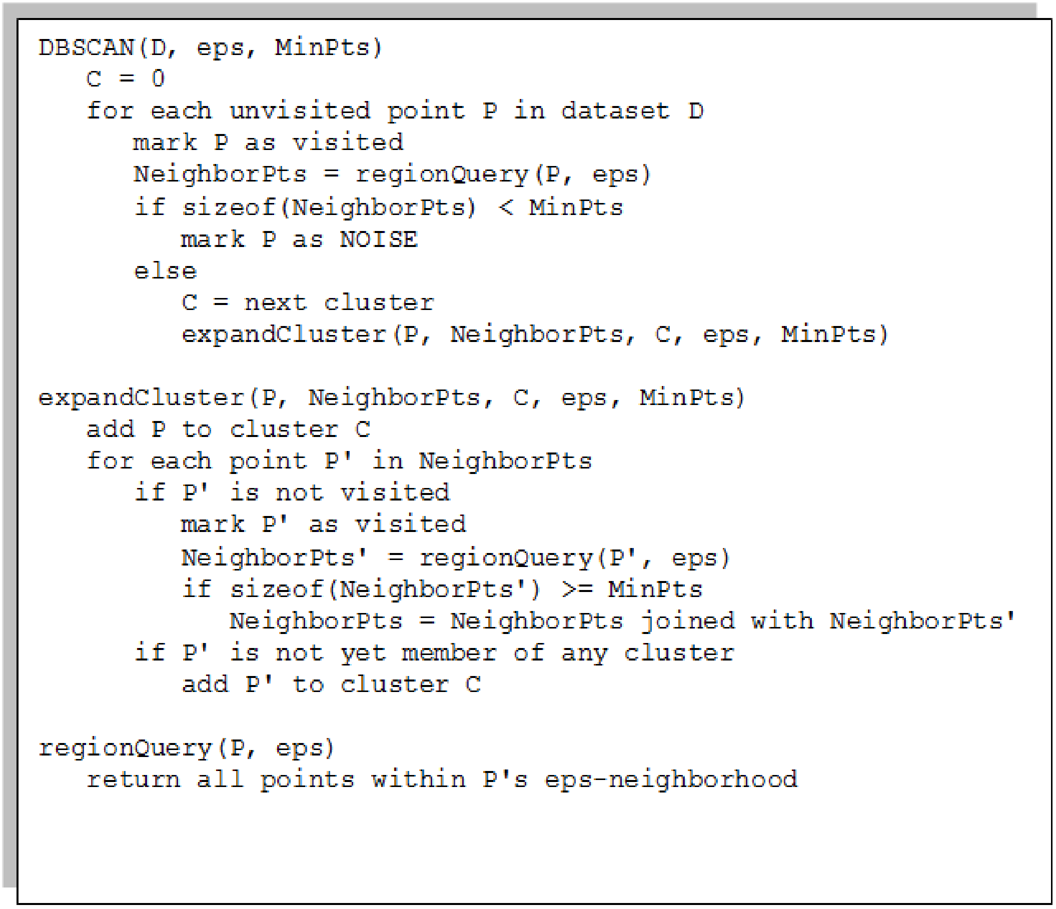
- Include as many samples of data as possible (the more data, the more accurate the results).
- To avoid distortions caused by excessive outliers, the median can be used instead of the mode.

6 DBSCAN Clustering Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a major data mining clustering method for Knowledge Discovery in Databases (KDD) which is used for unsupervised learning process[11]. In this paper we have used DBSCAN method for clustering the web page. Since web page contains large spatial set of data where there is no prior knowledge about the data set and also it is densely distributed so DBSCAN is implemented on it. DBSCAN algorithm has the ability in discovering clusters with arbitrary shape such as linear, concave, oval, etc. and it does not require the predetermination of the number of clusters[12]. DBSCAN has proven its ability of processing very large databases [11].

The DBSCAN algorithm takes two input parameters, epsilon (eps) or radius and the number of minimum points (minPts). The radius or eps is the distance between two data points to be considered as neighbors and minPts is the minimum required data points in a neighborhood to be considered a cluster. DBSCAN does not take as an input the number of clusters to generate, it finds the optimum number of clusters based on the minPts and eps[11]. Unlike K-means if a data point is not part of an existing cluster it is considered noise.[13]

6.1 The Algorithm[12]



```
DBSCAN(D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
   add P to cluster C
   for each point P' in NeighborPts
      if P' is not visited
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      if P' is not yet member of any cluster
         add P' to cluster C

regionQuery(P, eps)
   return all points within P's eps-neighborhood
```

6.2 Basic Concepts Of DBSCAN

DBSCAN is a density-based algorithm which is based on three concepts[12]:

1. **Core Point:** A point is considered to be a core point if it has more than a specified number of minPts within a radius or eps which are at the interior of a cluster.

2. **Border Point:** A point is known as border point if it has less than minPts within a radius or eps, and it needs to be in the neighborhood of a core point.

3. **Noise Point:** It is a point which is not a core point or a border point.

To consider cluster following things need to be considered :

- If any two **core points** are close enough within a distance **eps** of one another they are put in the **same cluster**

- If any **border point** that is close enough to a **core point** it is put **in the same cluster as the core point**

- All points that are found within the eps-neighborhood are added, as their own eps-neighborhood when they are dense

- This process continues until the density-connected cluster is completely found

- Other than these points that lie outside or unvisited node points are **noise points** which are discarded

6.3 Experimentation

We considered the web page structure as vector points. The web page contains numerous tags and from that tags we choose best 10 tags to classify our data in the cluster. These 10 tags is the representative of 10 dimensional vector that helped us to cluster the web pages via DBSCAN. We have tested the DBSCAN algorithm on three sets of web pages .The clustering technique is applied to cluster and distinguish three dissimilar sites which are Video, Sports and Forum site.

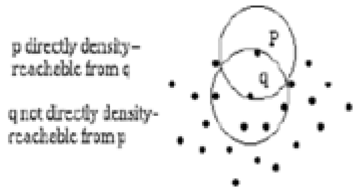
- SMALL (210 pages : less dense)
- MEDIUM (628 pages : medium dense)
- LARGE (3939 pages : largely dense)

The following section will describe further how the DBSCAN algorithm worked on the dataset that we have provided. The computing process is based on four rules or definitions:[12]

- **Rule1 (The eps-neighborhood of a point):** $NEps(p) = \{q \in D | dist(p,q) < Eps\}$ For a point to belong to a cluster it needs to have at least one other point that lies closer to it than the distance eps.
- **Rule 2: (Directly density-reachable):** The eps-neighborhood of a border point tends to have significantly less points than the eps-neighborhood of a core point.[11]

   1. The border points will still be a part of the cluster and in order to

include these points, they must belong to the eps-neighborhood of a core point q, i.e. $p \in NEps(q)$



Point *p* is directly density-reachable from point *q* but not vice versa.

2. In order for point q to be a core point it needs to have a minimum number of points within its eps-neighborhood. $|NEps(q)| \geq MinPts$ (core point condition)

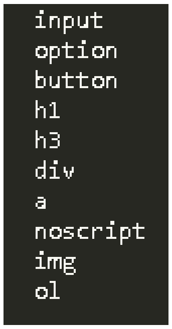
- **Rule 3 (cluster):** If point p is a part of a cluster C and point q is density-reachable from point p with respect to a given distance and a minimum number of points within that distance, then q is also a part of cluster C.
  $\forall p, q : p \in C$ and q is density-reachable from p with respect to eps and minPts, then $q \in C$.
  Therefore, to find a cluster, DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p with respect to eps and minPts. If p is a core point, this procedure yields a cluster with respect to eps and minPts. If p is a border point then no points are density-reachable from p and DBSCAN visits the next point of the database.

- **Rule 4 (noise):** Noise is the set of points, in the database, that dont belong to any of the clusters and they don't belong to any border or core point in eps-neighborhood.

The computation we have done is based on following the above rules. For every set of web pages we have considered different parameters of eps and minPts to get numerous clusters among the three websites (video, sports, forum). The estimation of the parameters is taken from the data set. We consider the three websites as three classes i.e. class 1 is video, class 2 is sport and class 3 is forum. For classify the web pages into clusters different tags are used (feature extraction). From which the DBSCAN algorithm has implemented 10 tags which is considered as 10 dimensional vector which represents web page structure.

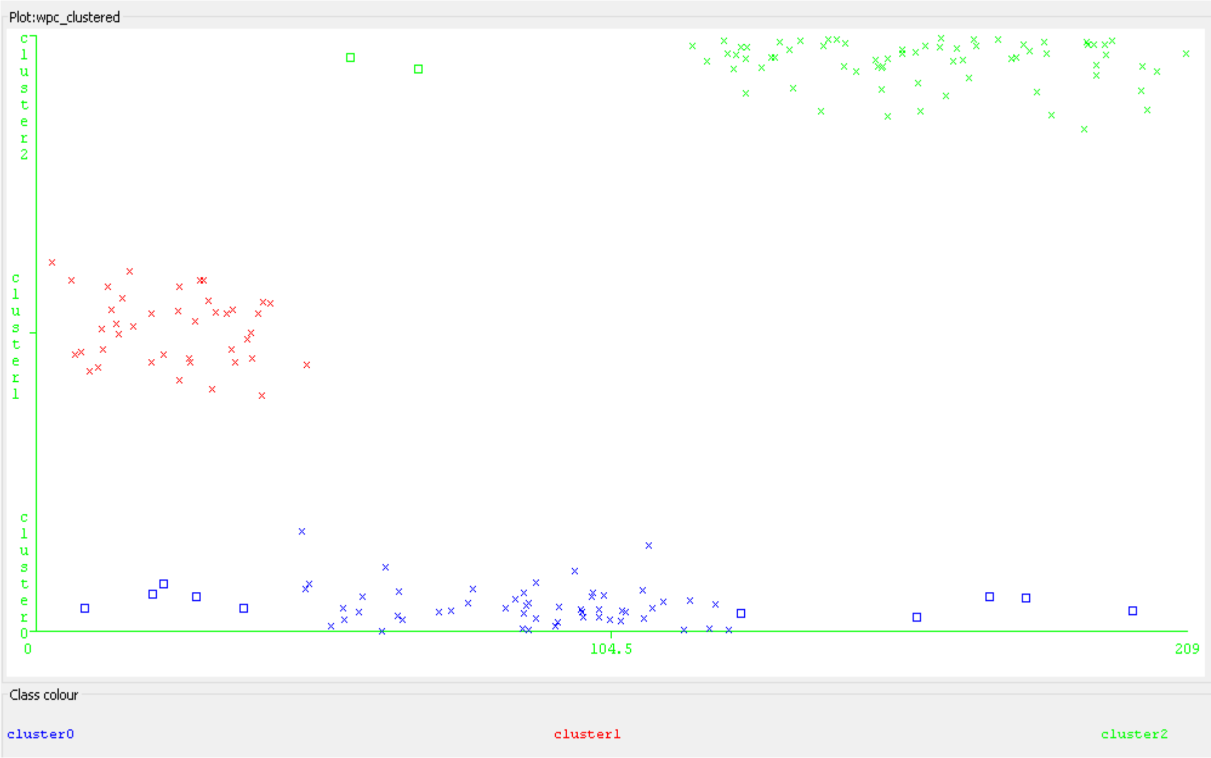
6.3.1 Small Set of Web pages (210)

The small set of web pages has data set of maximum range of radius 0.42. So we considered the average of the data set and taken the radius/eps = 0.4. The minPts = 10 is taken to determine clusters without much noise from sets of small data set of web pages. From the 10 vectors all the 10 vectors are chosen because small data set of web pages needs to be compared with every tags to distinguish web pages between the three sites (video, sport, forum).

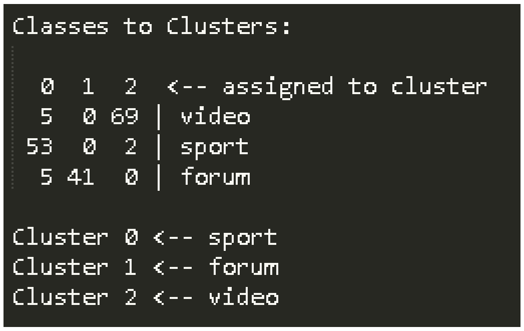


```
input
option
button
h1
h3
div
a
noscript
img
ol
```

Result: The DBSCAN algorithm is implemented over the 10 dimensional vector showed 94.2857% accuracy. The DBSCAN clustering resulted in forming three clusters. The results are shown below:



Here in the graph the blue color indicates sport, red color indicates forum and green color indicates video. The DBSCAN computes the core points and the border points at first and according to rule 1, $NEps(p) = \{q \in D|dist(p,q) < Eps\}$ that means the point p must be a point which is less than the distance from the eps radius and it is closer to the point that belongs to the cluster. So we can see that three clusters have been formed at three different positions that is seen from the y axis; cluster 0 indicates the blue color which means sport, cluster 1 indicates red color which means forum and cluster 2 indicates green color which means video.



```
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  5  0 69 | video
 53  0  2 | sport
  5 41  0 | forum

Cluster 0 <-- sport
Cluster 1 <-- forum
Cluster 2 <-- video
```

The table shows the cluster analysis of the web pages that has been performed by DBSCAN. Here are three clusters 0, 1, 2 which are sport, forum and video respectively. But here we can see some classes of sport, forum and video has incorrectly entered into the cluster of different category. The class video originally belongs to cluster 2 but has some points that have been incorrectly instanced. Here from the fig we can see at cluster 0 (the sport cluster) 5 video class has entered. Similarly 2 sport class has entered to cluster 2(the video cluster) and 5 forum class has entered to cluster 0. This happened because of rule 2 the point that has been incorrectly instanced are border points that are density reachable to the core point so they belong to the eps-neighborhood and thus they fall to the cluster that is of different category.

Here the total clustered instances are shown below:

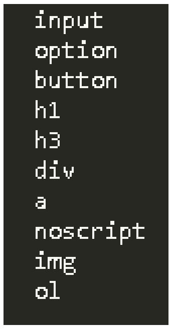
Cluster	Instances	Class
0	63 (36%)	Sport
1	41 (23%)	Forum
2	71 (41%)	Video

Here the incorrectly clustered instances are 12.0 (5.7143 %) which is almost negligible compared to success. The unclustered instances are 35 that means these are the instances that do not fall to any cluster and these are considered as noise points according to rule 4.

6.3.2 Medium Set of Web pages (628):

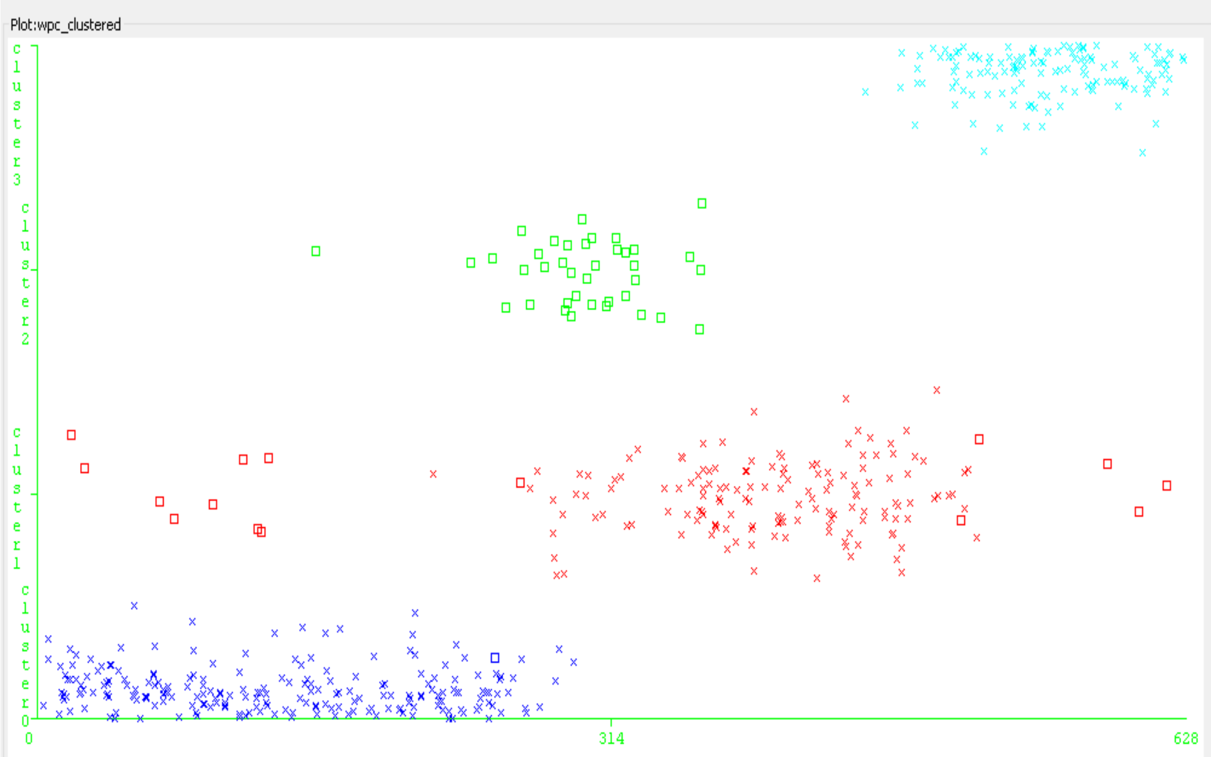
The medium set of web pages has data set of maximum range of radius 0.35. So we considered the average of the data set and taken the radius/eps = 0.29.The minPts = 15 is taken which is not more or not so less because if we have taken less minPts data set could have much noise from sets and if we have taken more then it may form the whole radius as a cluster. This is a balanced consideration for medium set of web pages.

From the 10 vectors all the 10 vectors are chosen because medium data set of web pages is also needs to be compared with every tags to distinguish web pages between the three sites (video, sport, forum).



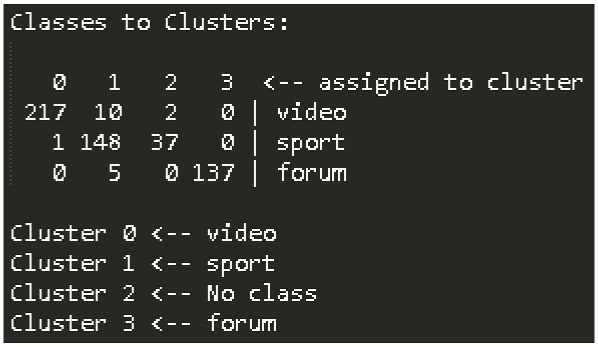
```
input
option
button
h1
h3
div
a
noscript
img
ol
```

Result: The DBSCAN algorithm is implemented over the 10 dimensional vector showed 91.256% accuracy. The DBSCAN clustering resulted in forming three clusters with one cluster that do not belong to any class. The results are shown below:



Here in the graph the blue color indicates video, red color indicates sport, light blue color indicates forum and green color indicates no class (that does not belong to any class).

The DBSCAN computes the core points and the border points at first and according to rule 1, $NEps(p) = \{q \in D|dist(p,q) < Eps\}$, that means the point p must be a point which is less than the distance from the eps radius and it is closer to the point that belongs to the cluster . So we can see that three clusters have been formed at three different positions that is seen from the y axis; cluster 0 indicates the blue color which means video, cluster 1 indicates red color which means sport and cluster 3 indicates light blue color which means forum. The green color that has form cluster does not belong to any class but it is a cluster because the points that have formed the cluster is still less than eps.



```
Classes to Clusters:

   0   1   2   3  <-- assigned to cluster
 217  10   2   0 | video
   1 148  37   0 | sport
   0   5   0 137 | forum

Cluster 0 <-- video
Cluster 1 <-- sport
Cluster 2 <-- No class
Cluster 3 <-- forum
```

From the second fig the table shows the cluster analysis of the web pages that has been performed by DBSCAN .Here are three clusters 0, 1, 3 which are video, sport and forum respectively. But here we can see some classes of sport, forum and video has incorrectly entered into the cluster of different category. The class video originally belongs to cluster 0 but has some points that have been incorrectly instanced. Here from the fig we can see at cluster 0 (the video cluster) 1 sport class has entered. Similarly 10 video class and 5 forum class has entered to cluster 1(the sport cluster). This happened because of rule 2 the point that has been incorrectly instanced are border points that are density reachable to the core point so they belong to the eps-neighborhood and thus they fall to the cluster that is of different category.

Here the total clustered instances are shown below:

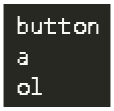
Cluster	Instances	Class
0	218 (39%)	Video
1	163 (29%)	Sport
2	137 (25%)	Forum

Here the incorrectly clustered instances are 55.0(8.744%) which is almost negligible compared to success. The unclustered instances are 72 that means these are the instances that do not fall to any cluster and these are considered as noise points according to rule 4. There is also a cluster that does not belong to any class which has instances 39(7%) which creates an inefficient result, because an extra cluster is not expected in clustering.

6.3.3 Large Set of Web pages (3939):

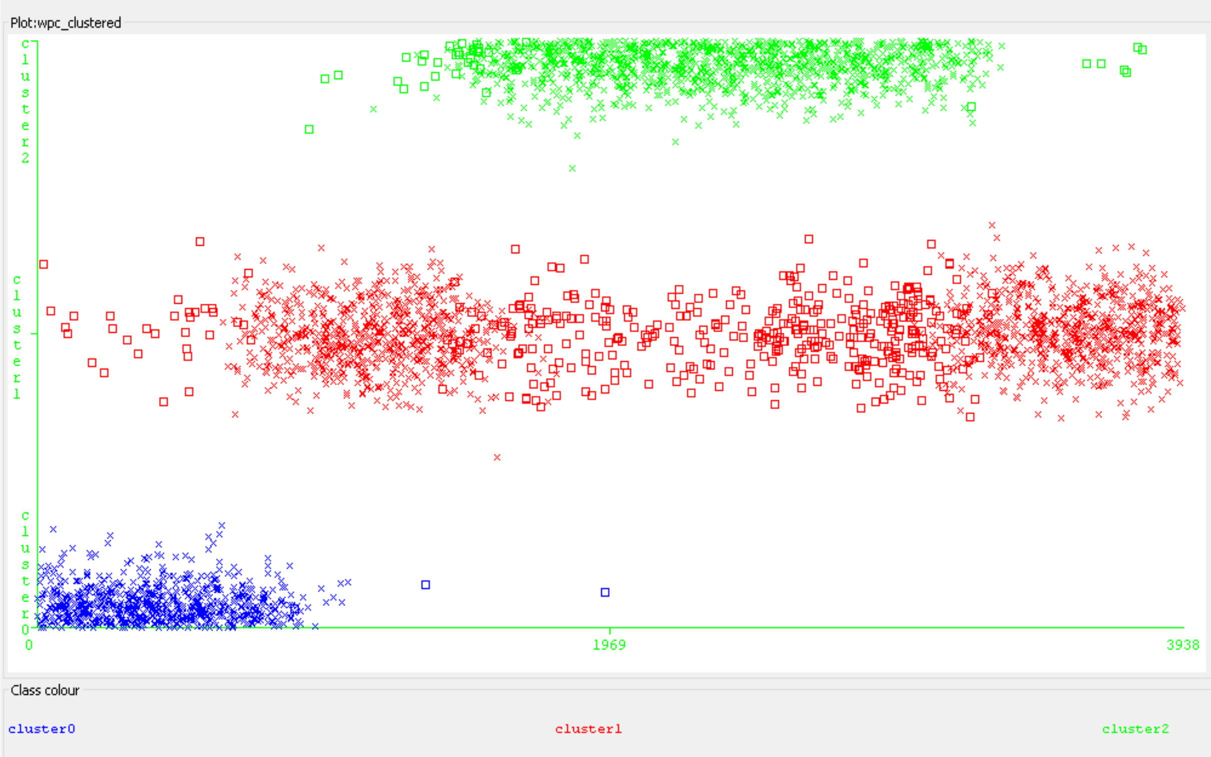
The large set of web pages has data set of maximum range of radius 0.1. So we considered the average of the data set and taken the radius /eps = 0.08.The minPts = 19 is taken which is more than medium and small set of web pages because here more minPts set has been taken so that we can avoid much noise from sets. This is a balanced consideration for large set of web pages.

From the 10 vectors only three vectors are chosen because large data set of web pages doesn’t need to be compared with every tags to distinguish web pages between the three sites (video, sport, forum) only major listed tags would help to make a clear distinction.



```
button
a
ol
```

Result: The DBSCAN algorithm is implemented over the 3 dimensional vector showed 88.8043% accuracy. The DBSCAN clustering resulted in forming three clusters with one cluster that do not belong to any class. The results are shown below:



Here in the graph the blue color indicates forum, red color indicates sport, green color indicates video.

The DBSCAN computes the core points and the border points at first and according to rule 1, $NEps(p) = \{q \in D | dist(p,q) < Eps\}$ that means the point p must be a point which is less than the distance from the eps radius and it is closer to the point that belongs to the cluster . So we can see that three clusters have been formed at three different positions that is seen from the y axis; cluster 0 indicates the blue color which means forum, cluster 1 indicates red color which means sport and cluster 3 indicates green color which means video.

From the second fig the table shows the cluster analysis of the web pages that has been performed by DBSCAN .Here are three clusters 0, 1, 2 which are forum, sport and video respectively. But here we can see some classes of video, sport and forum has incorrectly entered into the cluster of different category. The class forum originally belongs to cluster 0 but has some points that have been incorrectly instanced. Here from the fig we can see at cluster 0 (the forum cluster) 1 video and 2 sport class has entered. Similarly 368 video class and 32 forum class has entered to cluster 1(the sport cluster) and 38 sport class has entered to cluster 2(the video cluster). This happened because of rule 2 the point that has been incorrectly instanced are border points that are density reachable to the core point so they belong to the eps-neighborhood and thus they fall to the cluster that is of different category. But this result is not satisfactory compared to small and medium data set.

Here the total clustered instances are shown below:

Cluster	Instances	Class
0	750 (19%)	Forum
1	1857 (48%)	Sport
2	1265 (33%)	Video

Here the incorrectly clustered instances are 441.0(11.1957%) which is much lesser than medium and small set of web pages. The unclustered instances are 67 that means these are the instances that do not fall to any cluster and these are considered as noise points according to rule 4. This result is not so good compared to medium and small set, but compared to the large data set this result is satisfactory.

7 Conclusion

This thesis focused on one of the most fascinating aspect of modern web technology. We tried to find a new way to do web page clustering to add another way to apply this technology in more accurate and efficient way. K-means and dbscan have not been applied and compared directly to web clustering before. So this thesis work might have opened a new way to look at web page clustering. This new idea can be pursued by others by using more websites and by collecting more web pages from them. Our work has also shown a effective but simple way to represent web pages by using HTML tags. As the clustering algorithms were successful to cluster web pages with a satisfactory result, this representation is showing promise as another potential web page representation. We are proud to have contributed in the field of clustering web pages and to find some new ideas for future improvement in this field.

8 Future Work

Here we list some suggestions which can be followed if anyone wish to further pursue this thesis topic.

8.1 Distributed Crawler

The distributed crawler we implemented for this thesis is suitable for our purpose, but any further approach to better it may have benefit knowing some shortcomings of this crawler. Sections below offer some improvement criteria.

8.1.1 Avoid Single Point Of Failure

Our only database is residing on one single machine. If that machine dies there will be no database left for the database manager to contact to and all of the CrawlJobs will stop working. The whole crawling process will come to an halt. What could be done here is that, we could have some shadow databases in every machine which does not have the main database. After a certain period of time, all of these shadow databases will sync with the main database and update it to reflect the current state of the main database. In the case where the database machine fails. We can configure each database manager to connect to one of the available databases as fall back measure.

In this way even if all the machine dies except one, that only machine will be able to continue crawling by referring to its own local database.

8.1.2 Efficiency By Caching

Every database manager handles M number of CrawlJobs requests at any given time. That means it has to seek M number of unvisited web page URLs from the database. But each seek takes time as these are disk seeks. What we can do here is, we can pre-fetch M pages and keep those M pages in memory. So every for every M CrawlJobs we just have to do a single disk seek. This can increase the efficiency of the crawler by a factor of M.

References

[1] Shkapenyuk, Vladislav, and Torsten Suel. ”Design and implementation of a high-performance distributed web crawler.” Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE, 2002.

[2] Chakrabarti, Soumen, Martin Van den Berg, and Byron Dom. ”Focused crawling: a new approach to topic-specific Web resource discovery.” Computer Networks 31.11 (1999): 1623-1640.

[3] Brin, Sergey, and Lawrence Page. ”Reprint of: The anatomy of a large-scale hypertextual web search engine.” Computer networks 56.18 (2012): 3825-3833.

[4] Heydon, Allan, and Marc Najork. ”Mercator: A scalable, extensible web crawler.” World Wide Web 2.4 (1999): 219-229.

[5] Cunningham, P. ”Dimension Reduction, Technical report UCD-CSI-2007-7.” University College Dublin (2007).

[6] Data Mining Software In Java. Retrieved from http://www.cs.waikato.ac.nz/ml/weka/.

[7] Bellman, Richard, et al. Adaptive control processes: a guided tour. Vol. 4. Princeton: Princeton university press, 1961.

[8] Aggarwal, Charu C. ”A framework for clustering evolving data streams.” Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003.

[9] Basu, Sugato, Arindam Banerjee, and Raymond Mooney. ”Semi-supervised clustering by seeding.” In Proceedings of 19th International Conference on Machine Learning (ICML-2002. 2002.

[10] Kashima, Hisashi. ”K-means clustering of proportional data using L1 distance.” Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.

[11] Bacquet, Carlos. ”A comparison of unsupervised learning techniques for encrypted traffic identification.” Journal of Information Assurance and Security 5 (2010): 464-472.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Data Mining and Knowledge Discovery, pp226231, 1996.

[13] C.Sanjay, Prof. N.K.Nagwani ,D.Lopamudra.Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms. International Journal of Computer Applications (0975 8887), Vol. 27 No.11, pp14-18, 2011

[14] M. Oded and R. Lior .Data Mining and Knowledge Discovery Handbook, Springer.2010