Project C Report (Clustering Algorithms)

This report contains details about clustering algorithms. It mainly compares two of the clustering algorithms, K-Means and Buckshot and some variation of these algorithms. Clustering is performed on results obtained by Vector Space Similarity.

Overview of Clustering Algorithms

For a given query there can be results in multiple categories. E.g., for query "Michael Jordan" we can have some pages related to statistician and some pages related to basket ball player. User may be interested in any one of this. Clustering assists user to find only relevant document of his interest. Relevant documents are clustered using clustering. Good clustering should have high intra cluster similarity and low inter cluster similarity. Intra cluster similarity can be defined as,

$$S_a = \frac{1}{|C_i|} \sum_{d \in C_i} d \cdot c$$

where c is the centroid of the Ci'th cluster. And inter cluster similarity can be defined as,

$$S_e = \frac{1}{k * (k-1)} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} c_i \cdot c_j$$

where c_i and cj are centroids of i'th and j'th cluster respectively.

K Means Algorithm

In K Means algorithm initial K centroids are selected randomly. Then using Vector Space similarity each document is assigned to the cluster with maximum similarity. And the centroids are recomputed. Then all the data points are reassigned to the closest centroids. And this process repeats until there is any change in clusters in successive iterations.

The algorithm is as described below.

```
findClusters ( top N docs, k){
        for i = 1 to k{
                do{
                        randomly select a doc
                        cluster[i].centroid = selected doc.
                }while(doc not already selected)
        }
```

```
        set change = true
        While(change){
                assignClusters()
                computeCentroids()
        }
}

assignClusters(){
        set change = false
        for i = 1 to N{
                set maxSim = 0
                for j = 1 to k{
                        if sim(cluster [j].centroid , ith doc) > maxSim{
                                closestcluster = j
                                maxSim = sim(cluster [j].centroid , ith doc)
                        }
                }
                Assign ith doc to closest cluster
                if closestcluster  != previouscluster then
                        set change = true
        }
}

computeCentroids(){
        for i = 1 to k{
                set cluster[i].centroid = 0
                for each doc in cluster i{
                        cluster[i].centroid += doc
                }
                Cluster[i].centroid /= number of docs in cluster i
        }
}
```

Let **I**, **d**, **k** and **n** be the no. of iterations, no. of dimensions of a vector (average no. of terms), no. of clusters and no. of documents in base set respectively. Then the time complexity of assignClusters is of

O(nkd) and the time complexity of computeCentroids is O(nd).

So for I iterations time complexity becomes
O (k + I (nkd + nd)) = O (Idkn)

To store doc vector of top **n** documents space of O (nt) is required. Where **t** is the average no. of terms in a document. And to store centroid vector space of O (kt) is required, where **k** is the no. of centroids. And to assign cluster to each document space of O (n) is required.

So total space complexity is
O (nt + kt + n) = O(nt)

So this algorithm is low in cost. But its performance depends on the initial choice of centroids. Sometimes the results change dramatically during multiple runs of K Means algorithm because of randomly picking the initial centroids. Various extensions are applied on K Means to improve its performance.

Buckshot Algorithm

Buckshot algorithm tries to improve the performance of K Means algorithm by choosing better initial cluster centroids. For that it uses Hierarchical Agglomerative Clustering (HAC) algorithm. HAC considers each point as a separate cluster and combines the cluster with the maximum similarity. Similarity between clusters is measured as a group average. When the no. of clusters left equals the no. of required clusters, the algorithm is stopped. And these centroids of these clusters are taken as initial centroids for the K Means algorithm. For Buckshot algorithm HAC is performed on $\sqrt{kn}$ documents.

```
findInitialSeeds (top N docs, k) {
        create a clusterVector containing randomly selected √ kn  docs from top n docs

        while( clusterVector.size > k){
                maxSim = 0
                for i = 1to k{
                        for j = i + 1 to k{
                                if sim(clusterVector[i], clusterVector[j]) > maxSim{
                                        cluster1 = clusterVector[i]
                                        cluster2 = clusterVector[j]
                                        maxSim = sim(clusterVecor[i], clusterVector[j])
                                }
                        }
                }
                remove cluster1 and cluster2 from clusterVector
                combine docs of cluster1 and cluster2 into cluster
                add cluster to clusterVector
        }
        Return clusterVector
}
```

Let **n** be the number of documents to be clustered, **k** be the no. of clusters and **d** be the no. of dimensions.

Then time complexity of HAC is

$O(\sqrt{kn} + (\sqrt{kn} - k)\, knd) = O((kn)^{3/2}d)$

The time complexity of buckshot algorithm is O $((kn)^{3/2}d + Idkn)$.

And the space complexity of HAC is $(nt + kt + \sqrt{kn})$.
Where n is the total no. of documents, t is the average no. of terms per document and k is the no. of clusters.

So the space complexity of buckshot algorithm is O $(nt + kt + n + nt + kt + \sqrt{kn}) = O(nt)$

Bisecting K Means Algorithm

This algorithm tries to improve quality over K Means. It starts with one large cluster of all the data points and divides the whole dataset into two clusters. K Means algorithm is run multiple times to find a split that produce maximum intra cluster similarity. Then the cluster with largest size is picked to split further. This cluster can be chosen based upon minimum intra cluster similarity also. This algorithm is run k – 1 times to get k clusters.

This algorithm performs better than regular K Means because bisecting K Means produces almost uniform sized clusters. While in regular K Means there can be notable difference between sizes of the clusters. As small cluster tends to have high intra cluster similarity, large clusters have very low intra cluster similarity and overall intra cluster similarity decreases.
The algorithm is as described below.

```
findClusters ( top N docs, k){
        Initialize clusterVector with one cluster containing all documents
        for i = 1 to k - 1{
                find cluster with largest size from clusterVector
                remove cluster from clusterVector.

        Set maxsim = 0
        Do Iteration times{
                for j = 1 to 2{
                        do{
                                randomly select a doc
                                cluster[j].centroid = selected doc.
                        }while(doc not already selected)
                }

                set change = true
                While(change){
                        assignClusters()
                        computeCentroids()
                }
                if(sim > maxsim){
                        maxsim = sim
```

```
                    store clusters in cluster1 and cluster2
                }
        }
        Add cluster1 and cluster2 to clusterVector
}

assignClusters(){
        set change = false
        for i = 1 to N{
                set maxSim = 0
                for j = 1 to 2{
                        if sim(cluster [j].centroid , ith doc) > maxSim{
                                closestcluster = j
                                maxSim = sim(cluster [j].centroid , ith doc)
                        }
                }
                Assign ith doc to closest cluster
                if closestcluster  != previouscluster then
                        set change = true
        }
}

computeCentroids(){
        for i = 1 to 2{
                set cluster[i].centroid = 0
                for each doc in cluster i{
                        cluster[i].centroid += doc
                }
                Cluster[i].centroid /= number of docs in cluster i
        }
}
```

Let **n** be the number of documents to be clustered, **k** be the no. of clusters, **i** be the no. of iterations made on each run and **d** be the no. of dimensions.

Then the time complexity of assignClusters is O (nd). And time complexity of computeCentroids is O (nd) if average size of cluster is of O (n).

So overall complexity of bisecting K Means algorithm is
O ((k - 1) * i * (nd + nd)) = O (nkid)

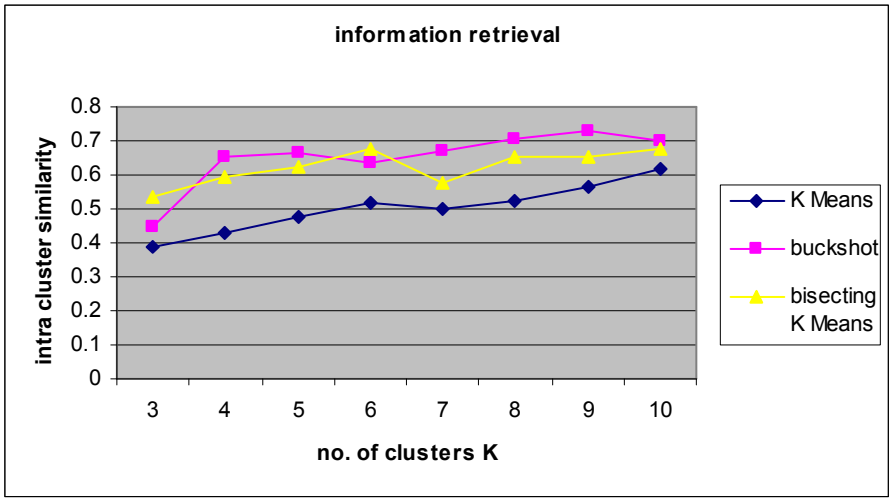
So time complexity of bisecting K Means is linear in nature with respect to no. of documents.
To store doc vector of top **n** documents space of O (nt) is required. Where **t** is the average no. of terms in a document. And to store centroid vector space of O (kt) is required,

where **k** is the no. of centroids. And to assign cluster to each document space of O (n) is required.So total space complexity is
O (nt + kt + n) = O(nt)

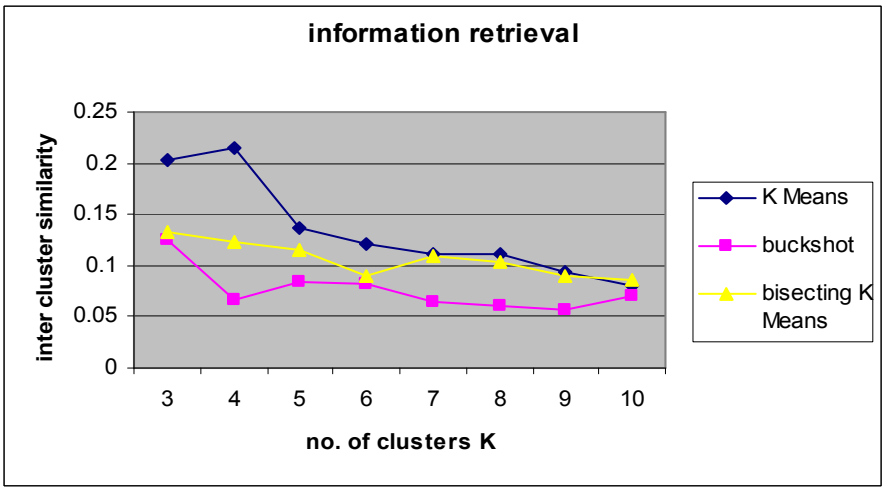
This algorithm is low in cost compare to buckshot algorithm but still gives comparable performance to buckshot algorithm.

Comparison of algorithms using similarity measures

To compare these algorithms inter cluster and intra cluster similarity measures are used. Algorithms were run for k = 3 to k =10 for 2 queries "information retrieval" and "parking decal".

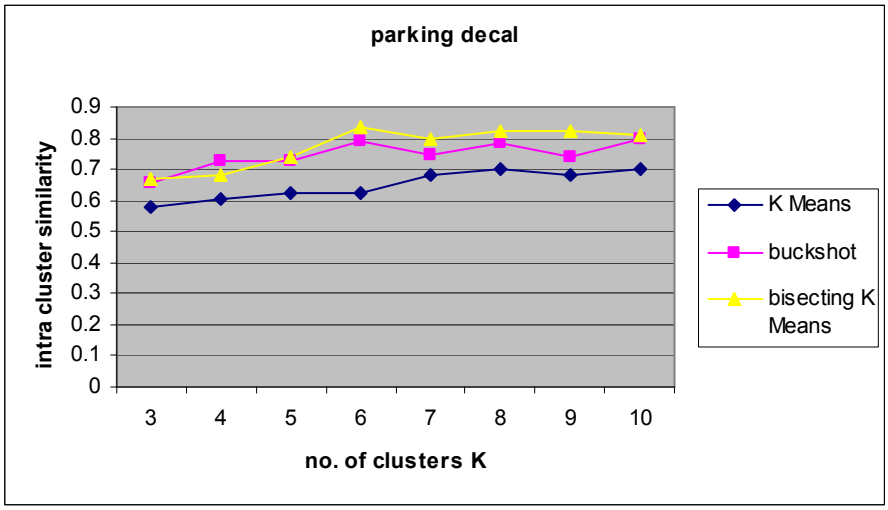


Intra Cluster Similarity for all algorithms on "Information retrieval" query

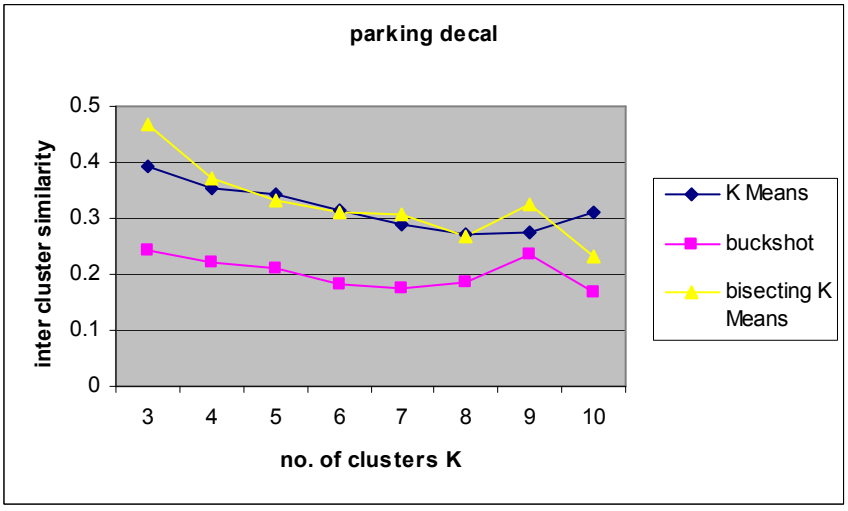


Inter Cluster Similarity for all algorithms on "Information retrieval" query

An algorithm performs well compared to another one if it has higher intra cluster similarity and lower inter cluster similarity compared to other algorithm. From the graphs it is obvious that buckshot algorithm consistently performs better than other algorithms in both intra cluster and inter cluster similarity measure. For k =3 and 6 bisecting K Means performs slightly better than buckshot algorithm in intra cluster similarity measure but overall buckshot algorithm outperforms other algorithms.



Intra Cluster Similarity for all algorithms on "Parking decal" query



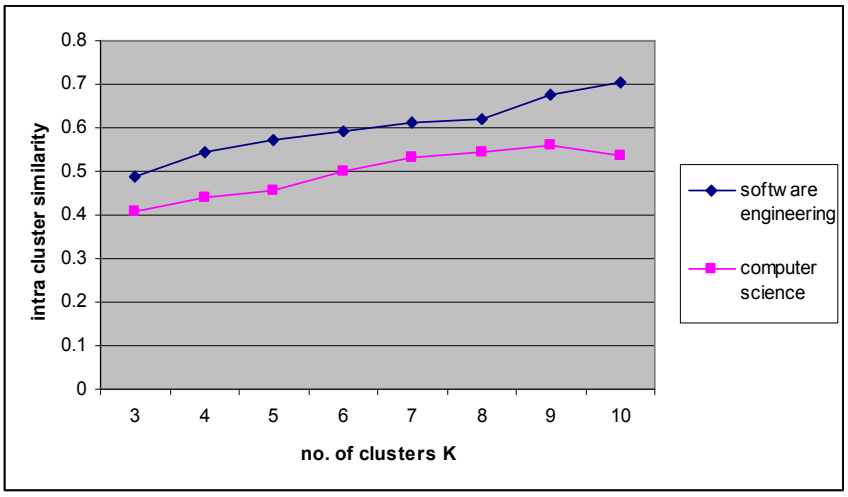
Inter Cluster Similarity for all algorithms on "Parking Decal" query

From the 1st graph it is clear that buckshot and bisecting K Means algorithm almost perform equal in terms of intra cluster similarity. And it is already mentioned that buckshot algorithm performs well because it selects better initial centroids. Bisecting K

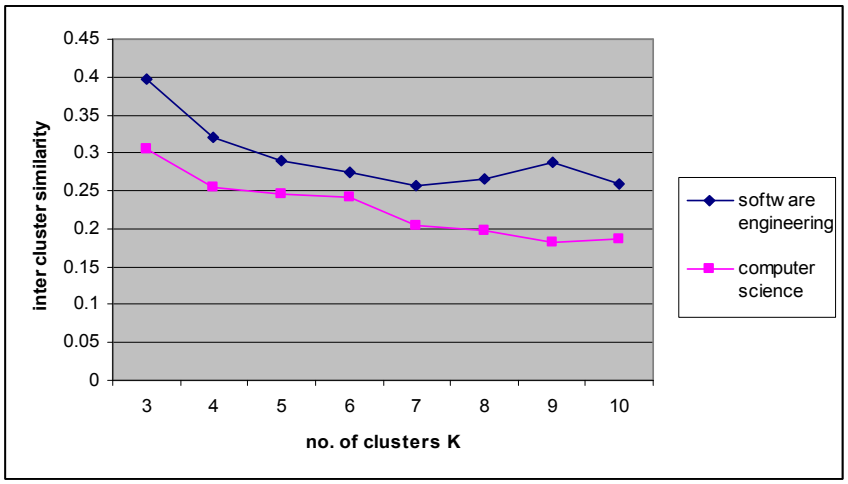
Means algorithm performs well compared to K Means because Bisecting K Means algorithm produces clusters of almost uniform sizes.

Effect of varying no. of clusters

To analyze the effect of varying no. of clusters results of intra cluster similarity and inter cluster similarity for two queries for K Means algorithm are presented.



Intra cluster similarity for 2 queries for K Means algorithm



Inter cluster similarity for 2 queries for K Means algorithm

From the 1st graph it is obvious that as the no. of clusters increases intra cluster similarity increases. And from 2nd graph it is obvious that as the no. of clusters increases inter cluster similarity decreases. This is due to the fact that as the no. of cluster increases the

cluster size becomes smaller so intra cluster similarity increases. And similarly inter cluster similarity decreases.

Analysis of clusters with respect to natural category

information retrieval

Cluster 1
www.public.asu.edu%%~hdavulcu%%CSE591_Semantic_Web_Mining.html
www.public.asu.edu%%~hdavulcu%%CSE591_Semantic_Web_Mining2.html
prism.asu.edu%%resources_links.asp

Cluster 2
www.asu.edu%%copp%%justice%%courses%%index.htm
www.asu.edu%%clas%%psych%%dinfo%%courses.htm

Cluster 3
www.eas.asu.edu%%~csedept%%academic%%courses.shtml
www.eas.asu.edu%%~csedept%%academic%%courses_pfv.html
www.asu.edu%%aad%%catalogs%%spring_2003%%cse.html

Cluster 4
www.public.asu.edu%%~candan%%cv.htm
www.public.asu.edu%%~candan%%research.htm
aria.asu.edu%%people.htm

Cluster 5
www.eas.asu.edu%%~gcss%%people%%nvf%%pubs.html
wpcarey.asu.edu%%pubs%%is%%pub.cfm
www.eas.asu.edu%%~gcss%%people%%nvf%%vita.html

Above given are the results for query information retrieval using K Means algorithm using k = 5. 1st cluster represents semantic web mining course pages. 2nd cluster represents pages related to courses. Cluster 3 represents cse courses page. Cluster 4 represents pages from prof. candan's directory. And cluster 5 has pages from gcss. So almost all clusters have pages forming one category.

Computer Science

Cluster1
www.eas.asu.edu%%~wcs%%index.html
www.eas.asu.edu%%~wcs%%events.htm
www.eas.asu.edu%%~wcs%%members.htm

Cluster 2
www.asu.edu%%provost%%smis%%ceas%%bse%%csebse.html
www.asu.edu%%provost%%smis%%ceas%%bs%%csbs.html
www.eas.asu.edu%%~gcss%%people%%nvf%%pubs.html

Cluster 3
www.asu.edu%%lib%%noble%%library%%bestind.htm
www.asu.edu%%provost%%smis%%clas%%bs%%psbs.html
www.asu.edu%%provost%%smis%%clas%%ba%%psba.html

Above given are the results for computer science query using buckshot algorithm with K = 3. Cluster 1 represents pages from **wcs** directory. Cluster2 represents pages under **provost** directory. Cluster 3 represents pages from **clas** directory. So results for buckshot algorithm are also producing clusters which can be named.

Computer Science

Cluster 1
www.asu.edu%%lrc%%computerlab.html
www.asu.edu%%vpsa%%lrc%%computerlab.html

Cluster 2
www.eas.asu.edu%%~csedept%%Students%%Internships%%internships.shtml
www.eas.asu.edu%%~csedept%%Students%%Scholarships%%scholarships.shtml
www.eas.asu.edu%%~csedept%%AcademicPrograms%%AcademicPrograms.shtml

Cluster 3
www.eas.asu.edu%%~gcss%%people%%nvf%%pubs.html
www.asu.edu%%provost%%smis%%ceas%%bs%%csbs.html
www.asu.edu%%provost%%smis%%ceas%%bse%%csebse.html

Above are the results for computer science query using bisecting K Means algorithm for K = 3. Cluster 1 represents pages related to **computer lab**. Cluster 2 represents pages related to **scholarships.** Cluster 3 represents pages under **ceas** directory. Results for bisecting K Means are also producing clusters that can be named.