Empirical Comparison of Performances of K-Means, K-Means++, Weighted K-Means and Hartigan and Wong K-Means Clustering Algorithms

Technical Report · October 2015

DOI: 10.13140/RG.2.1.4481.7364

Ankit Choudhary
Indian Institute of Engineering Science and Technology, Shibpur

Empirical Comparison of Performances of K-Means, K-Means++, Weighted K-Means and Hartigan and Wong K-Means Clustering Algorithms

***Abstract*—K-Means is a popular and widely used unsupervised machine learning algorithm having several interesting variations. This paper empirically compares K-Means algorithm with its variations- K-Means++, Weighted K-Means and Hartigan and Wong K-Means. The comparison uses both internal and external performance indices on a wide variety of real life datasets from different domains. It has been found that density based Hartigan and Wong K-Means algorithm performs best. In this light future direction of research is suggested.**

Index Terms—***machine learning; unsupervised learning (clustering); distance measure; K-Means algorithm; internal and external validity indices.***

1. Introduction

Machine learning deals with automatic elicitation of knowledge from records of solved cases. If the records of solved cases have their classes defined by the expert then it is called supervised learning (classification) problem. However in real life classifications by experts are not always available. Learning from such data is known as unsupervised learning (clustering). In clustering problem one has to group several records together in one cluster and thus create different clusters from a set of records of solved cases. Each cluster can then be assigned to a particular class. There are several well-known clustering algorithms [1] which can be broadly classified as partitional and hierarchical. Among the partitional algorithms, K-Means clustering [2], [3] is most popular and well researched with several interesting variations [4], [5], [6].

This paper deals with empirical comparisons of three variations of K-Means clustering algorithm. Section 2 discusses K-Means algorithm and its different variations on which the empirical comparison is carried out. Section 3 deals with the empirical evaluation by defining the performance indices on which the comparisons are made. It also includes the experimental results and subsequent discussion on the results. The final section concludes the paper by setting the direction of future work in the light of empirical comparisons.

2. Background

The clustering problem statement can be defined as the task to *organize a given set of data points into k clusters.* The given dataset have *n* points in a *d* dimensional space, *A*. They have to be clustered into *k* distinct clusters.

A. K-Means Clustering

K-Means Algorithm (KMA), shown in Table I, is one of the most popular and most common clustering algorithms [2] [3]. It's a greedy algorithm that guaranties convergence to local minima. However it is known to be NP-Hard [7].

TABLE I. K-MEANS ALGORITHM (KMA)

1) *Choose k (signifying number of clusters) random points from dataset points x_i, where i= 1 to n, as initial centroids c_k..*
2) *For each point x_i, calculate the Euclidean distance from the centroids c_k of each clusters using*

   $$dist= \sqrt{\sum_{i=1}^{d}(c_i - x_i)^2}\ , \qquad (1)$$

   for d dimensional attribute space.
3) *Assign each point x_i to the nearest centroid cluster C_k based on its Euclidean distance.*
4) *For all clusters k, update the cluster centroids c_k using*

   $$c_k = \frac{\sum_{x_i \in C_k} x_i}{|C_k|} \qquad (2)$$

5) *Compute Sum of Squared Errors*

   $$SSE(C_i) = \sum_{i=1}^{k}\sum_{x \in C_i} dist^2(ci, x) \qquad (3)$$

6) *Repeat Step 2 to Step 5 until the convergence criterion is met.*

The convergence criteria for K-Means clustering and its various forms is that no point should change its cluster in the next iteration or there is an increase in SSE in an iteration compared to its previous one.

B. K-Means++

K-Means++ Algorithm [4] carefully chooses initial centroids for K-Means clustering to reduce bad approximation found in K-Means algorithm with respect to objective function (SSE) compared to optimal clustering. To this end it follows a simple probabilistic approach that starts by taking initially the first centroid randomly and the rest uniformly based on weighted probability score. This means that the second centroid selected is the one with probability proportional to *dist*^{2}, which is the farthest from the first centroid and so on. The complete algorithm is shown in Table II. The convergence criteria is same as that of KMA.

TABLE II. K-MEANS++ ALGORITHM

1) *Choose the* 1^{st} *initial centroid randomly among the dataset points,* x_i.
2) *Repeat*
3) *Choose the next centroid, using equation (1), farthest from the previously chosen centroid.*
4) *Until all initial k centroids are chosen.*
5) *For each point* x_i, *calculate the Euclidean distance, using equation (1), from the centroids* c_i *of k clusters* C_k.
6) *Assign each point to the nearest centroid cluster.*
7) *Update the cluster centroids for all the clusters using equation (2).*
8) Compute $SSE(C_i)$ *using equation (3).*
9) *Repeat Step 5 to Step 8 till the convergence criterion is met.*

C. Weighted K-Means

The distance measure of Weighted K-Means Algorithm [5] is weighted by the feature weights. That means standard K-Means Algorithm is augmented with a feature weighting mechanism. The weights for different features are learned automatically in this iterative optimisation algorithm. The optimisation function SSE is modified as shown in equation (7) where feature vector is numbered from v=1,…,d, the clusters are numbered from K=1,…,k and β is a user defined parameter to determine impact of the feature weights on the clustering. Equation (6) is used to update feature weights w_v. The convergence criterion for Weighted K-Means Algorithm is same as K-Means Algorithm and suffers similar issues of convergence as that of K-Means. The complete algorithm is shown in Table III.

TABLE III. WEIGHTED K-MEANS ALGORITHM

1) *Choose k initial centroids randomly and initialize the weights, w, for d dimensions such that their sum is 1, i.e.*

$$w_v = \frac{1}{d}, \forall\, v = 1 \text{ to } d \qquad (4)$$

$$\sum_{v=1}^{d} w_v = 1 \qquad (5)$$

2) *Assign each point* x_i *to a cluster* C_K *whose centroid* c_i *is closest to it using distance formula:*

$$d(x_i, c_K) = \sum_{v=1}^{d} w_v{}^{\beta}\,(x_{iv} - c_{Kv})^2, \qquad (5)$$

where β is a user defined variable which has an impact on the weights w_v.

3) *Update the weights of all the dimensions using*

$$w_v = \frac{1}{\sum_{u \in V} |\frac{D_v}{D_u}|^{\frac{1}{\beta-1}}}, \qquad (6)$$

where D_u *is the sum of inter-cluster variances of feature v weighted by cluster cardinalities* (D_v).

4) *Update the centroids of the clusters using equation (2).*
5) *Calculate the* $SSE(C_k, w)$ *using*

$$SSE(C_k, w) = \sum_{K=1}^{k} \sum_{x_i \in C_K} \sum_{v=1}^{d} s_{xiK}\, w_v{}^{\beta} (x_{iv} - c_{Kv})^2, \qquad (7)$$

where c_{Kv} *is the vth dimension of Kth cluster centroid and*

$$s_{xiK} \in (0, 1) \qquad (8)$$

$$\sum_{K=1}^{k} s_{xiK} = 1 \qquad (9)$$

i.e., s_{xiK} *is 1 if a point belongs to a cluster and 0 if it does not belong to a cluster, for hard clustered dataset.*

6) *Repeat Step2 to Step5 till the convergence criteria has been fulfilled.*

D. Hartigan and Wong K-Means

Hartigan and Wong [6] suggested that the points which had large number of points surrounding them within a hyper-sphere and are well separated from themselves are good candidates for initial centroids. Thus, their algorithm is based on the concept of nearest neighbour density within the hyper-sphere of a fixed radius *r*. To determine the density, first the pair wise Euclidean distance between points is computed. Then, for each point the density of its surrounding hyper-sphere is calculated using equation (10). Next, points are sorted in order of their decreasing density. The first point from the sorted array is chosen as the first initial centroid c_1. Subsequent initial centroids (k-1) are chosen maintaining a user specified distance *l* from the sorted array. The rest of the algorithm is similar to the K-Means Algorithm including its convergence criteria. The algorithm is shown in Table IV.

TABLE IV. HARTIGAN AND WONG ALGORITHM

1) *Calculate the densities of each point at a radius r, given by the user, and sort them according to their densities in descending order.*

$$Density(x) = \frac{Number\ of\ points\ within\ distance\ r\ from\ x_i}{r^d} \qquad (10)$$

2) *Choose the point with the highest density as the first initial centroid.*
3) *Choose the next highest density point at a distance of l (a user defined variable) or more from the last chosen centroid as the next centroid using equation (1).*
4) *Repeat Step3 till number of initial centroids equal k.*

5) *For each point x_i, calculate the Euclidean distance, using equation (1), from the centroids c_i of k clusters C_k.*
6) *Assign the points to clusters with the closest centroids.*
7) *Calculate the new centroids of the clusters using equation (2).*
8) *Compute $SSE(C_i)$ using equation (3).*
9) *Repeat Step 5 to 8 till the convergence criteria is met.*

3. Empirical Evaluation

The algorithms mentioned above are initially implemented on the 20 datasets used for the study with the number of clusters (k) as the original one for these datasets. The performance indices' for all the clustered files are calculated. Then, the datasets are clustered with value of k incremented and decremented. The performance indices' for these are calculated. If there is an improvement in the performance indices' for either or both of the changed k values, the process is repeated till we get the value of k corresponding to the best clustering.

A. Datasets

For this study, 20 real-life datasets corresponding to various domains like medical science, biology, marketing, political etc. have been taken. These datasets are clustered using the clustering algorithms mentioned above. The datasets have been taken from the UCI machine learning data repository [8]. They have varying sizes and attributes.

B. Performance Indices

Since clustering is an unsupervised problem, there is no way to check if the grouping of data points to clusters (using clustering algorithm) is correct or not. Hence, different measures called *Validity Indices* are used to test cluster quality. These indices can also be used to compare how well different clustering algorithms perform on the same dataset. Here we have used two types of indices: External and Internal indices [9], [10] for cluster evaluation. *External indices* evaluate the clustering based on a pre-specified clustering structure. On the other hand, *Internal Indices* evaluate cluster quality based on internal criteria.

Now we briefly define the indices used in our evaluation:

1) External Indices

For the previous stated dataset X, B = {B_1,…,B_q} is a defined partition of the dataset and the partition obtained from algorithm is C = {C_1,…,C_k}. Considering pair of points in the dataset, the following terms are defined as:

True Positive, **TP** - Number of pairs of points in X that are in the same set in C and in the same set in B.

True Negative, **TN** - Number of pairs of points in X that are in different sets in C and in different sets in B.

False Positive, **FP** - Number of pairs of points in X that are in the same set in C but in different sets in B.

False Negative, **FN** - Number of pairs of points in X that are in different sets in C but in the same set in B.

The external indices can be defined using the above stated terms as follows:

a) Rand Index:

$$(TP + TN) / (TP + FP + TP + TN) \qquad (11)$$

b) Jaccard Index:

$$TP / (TP + FP + FN) \qquad (12)$$

c) Fowlkes–Mallows index:

$$FM = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}} \qquad (13)$$

These external indices can be used to compare clustering structure C with structure B.

2) Internal Indices

a) Silhouette Index

Silhouette Index combines ideas of both compaction and separation, but for individual points, as well as clusters. The average silhouette coefficient over all the clusters is a measure of how well the dataset has been partitioned. Its value lies between -1 and 1. Higher value indicates better cluster.

b) Dunn Index

Dunn Index finds compact and well separated clusters. It can be defined as:

$$\mathrm{D} = \min_{1 \le i \le k} \left\{ \min_{1 \le j \le k,\, i \ne j} \left\{ \frac{d(i,j)}{\max_{1 \le l \le k} d'(l)} \right\} \right\} \qquad (14)$$

Where $d(i,j)$ is the distance between cluster i and j, and $d'(l)$ is the intra-cluster distance of cluster l. From equation (14) it can be concluded that larger the value of the index more compact and well separated are the clusters.

c) Davies-Bouldin Index

Davies-Bouldin Index, DB index, can be defined as:

$$\mathrm{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \le j \le k,\, i \ne j} \left\{ \frac{\mathrm{diam}(i) + \mathrm{diam}(j)}{d(i,j)} \right\} \qquad (15)$$

Where *diam(i)* and *diam(j)* are the diameters of cluster i and j.

Equation (15) indicates that lower value of DB index identifies compact and well separated clusters.

C. Experimental Results

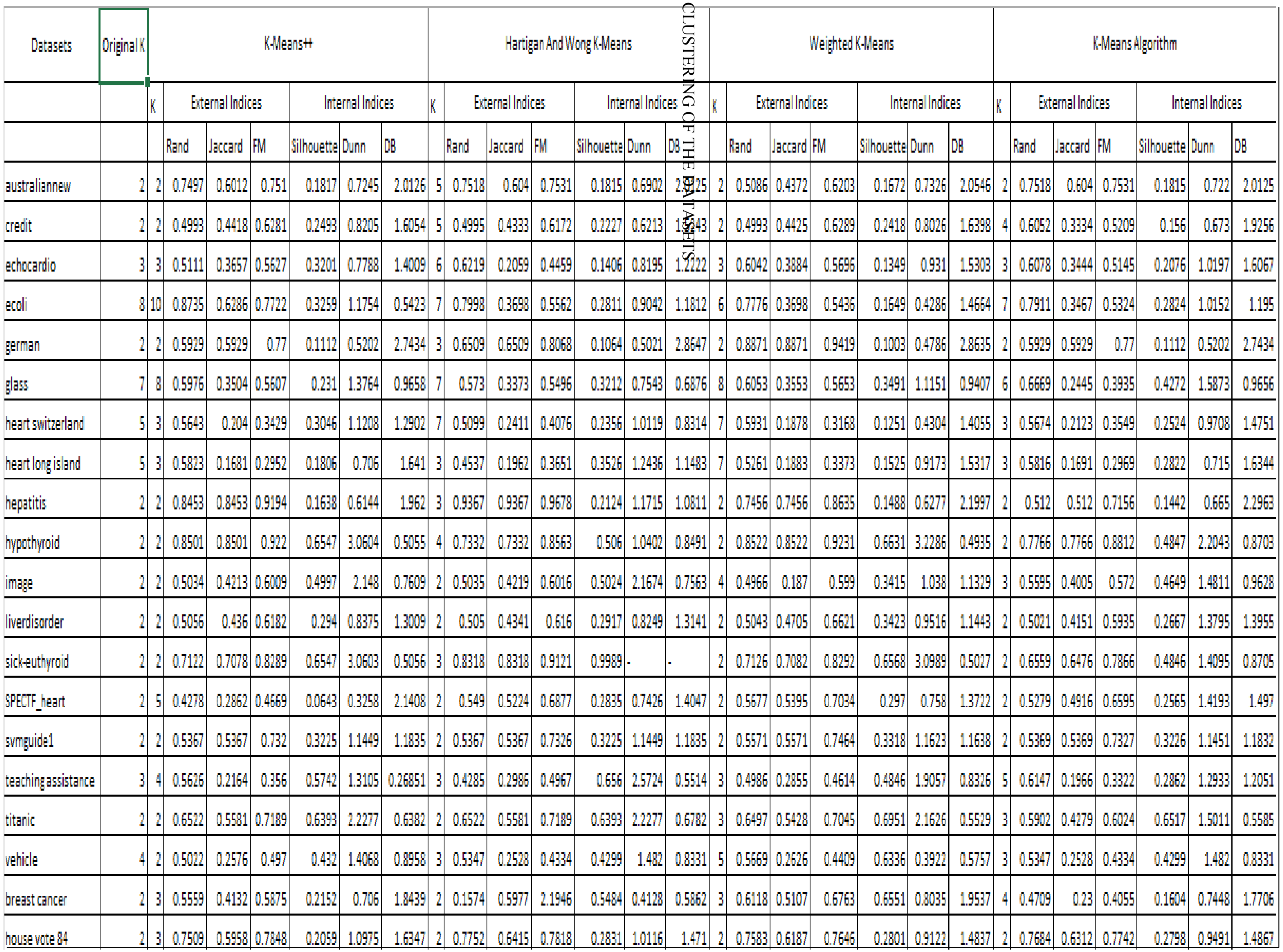
Table V shows the results of the clustering on the 20 datasets using the four clustering algorithms detailed in Section 2 based on the 6 performance indices. Table VI shows the average values of the performance indices for the 4 clustering algorithms over all the datasets and a score card based on these values. The clustering algorithm with the best value of a performance index gets a score of 4, second-best gets 3 and so on. In this way, the 4 algorithms are scored for six performance indices and the sums of their ranks are shown in Table VI.

D. Discussions

As it can be seen from Table VI, all the algorithms are better than the conventional K-Means algorithm.

The K-Means++ algorithm is best among the four with respect to Dunn Index. However, with respect to Rank Score it

TAL RESULTS FOR THE CLUSTERING OF THE DATASETS



Datasets	Original K	K-Means++							Hartigan And Wong K-Means							Weighted K-Means							K-Means Algorithm						
		K	External Indices			Internal Indices			K	External Indices			Internal Indices			K	External Indices			Internal Indices			K	External Indices			Internal Indices		
			Rand	Jaccard	FM	Silhouette	Dunn	DB		Rand	Jaccard	FM	Silhouette	Dunn	DB		Rand	Jaccard	FM	Silhouette	Dunn	DB		Rand	Jaccard	FM	Silhouette	Dunn	DB
australiannew	2	2	0.7497	0.6012	0.751	0.1817	0.7245	2.0126	5	0.7518	0.604	0.7531	0.1815	0.6902	2.0125	2	0.5086	0.4372	0.6203	0.1672	0.7326	2.0546	2	0.7518	0.604	0.7531	0.1815	0.722	2.0125
credit	2	2	0.4993	0.4418	0.6281	0.2493	0.8205	1.6054	5	0.4995	0.4333	0.6172	0.2227	0.6213	1.5343	2	0.4993	0.4425	0.6289	0.2418	0.8026	1.6398	4	0.6052	0.3334	0.5209	0.156	0.673	1.9256
echocardio	3	3	0.5111	0.3657	0.5627	0.3201	0.7788	1.4009	6	0.6219	0.2059	0.4459	0.1406	0.8195	1.2222	3	0.6042	0.3884	0.5696	0.1349	0.931	1.5303	3	0.6078	0.3444	0.5145	0.2076	1.0197	1.6067
ecoli	8	10	0.8735	0.6286	0.7722	0.3259	1.1754	0.5423	7	0.7998	0.3698	0.5562	0.2811	0.9042	1.1812	6	0.7776	0.3698	0.5436	0.1649	0.4286	1.4664	7	0.7911	0.3467	0.5324	0.2824	1.0152	1.195
german	2	2	0.5929	0.5929	0.77	0.1112	0.5202	2.7434	3	0.6509	0.6509	0.8068	0.1064	0.5021	2.8647	2	0.8871	0.8871	0.9419	0.1003	0.4786	2.8635	2	0.5929	0.5929	0.77	0.1112	0.5202	2.7434
glass	7	8	0.5976	0.3504	0.5607	0.231	1.3764	0.9658	7	0.573	0.3373	0.5496	0.3212	0.7543	0.6876	8	0.6053	0.3553	0.5653	0.3491	1.1151	0.9407	6	0.6669	0.2445	0.3935	0.4272	1.5873	0.9656
heart switzerland	5	3	0.5643	0.204	0.3429	0.3046	1.1208	1.2902	7	0.5099	0.2411	0.4076	0.2356	1.0119	0.8314	7	0.5931	0.1878	0.3168	0.1251	0.4304	1.4055	3	0.5674	0.2123	0.3549	0.2524	0.9708	1.4751
heart long island	5	3	0.5823	0.1681	0.2952	0.1806	0.706	1.641	3	0.4537	0.1962	0.3651	0.3526	1.2436	1.1483	7	0.5261	0.1883	0.3373	0.1525	0.9173	1.5317	3	0.5816	0.1691	0.2969	0.2822	0.715	1.6344
hepatitis	2	2	0.8453	0.8453	0.9194	0.1638	0.6144	1.962	3	0.9367	0.9367	0.9678	0.2124	1.1715	1.0811	2	0.7456	0.7456	0.8635	0.1488	0.6277	2.1997	2	0.512	0.512	0.7156	0.1442	0.665	2.2963
hypothyroid	2	2	0.8501	0.8501	0.922	0.6547	3.0604	0.5055	4	0.7332	0.7332	0.8563	0.506	1.0402	0.8491	2	0.8522	0.8522	0.9231	0.6631	3.2286	0.4935	2	0.7766	0.7766	0.8812	0.4847	2.2043	0.8703
image	2	2	0.5034	0.4213	0.6009	0.4997	2.148	0.7609	2	0.5035	0.4219	0.6016	0.5024	2.1674	0.7563	4	0.4966	0.187	0.599	0.3415	1.038	1.1329	3	0.5595	0.4005	0.572	0.4649	1.4811	0.9628
liverdisorder	2	2	0.5056	0.436	0.6182	0.294	0.8375	1.3009	2	0.505	0.4341	0.616	0.2917	0.8249	1.3141	2	0.5043	0.4705	0.6621	0.3423	0.9516	1.1443	2	0.5021	0.4151	0.5935	0.2667	1.3795	1.3955
sick-euthyroid	2	2	0.7122	0.7078	0.8289	0.6547	3.0603	0.5056	3	0.8318	0.8318	0.9121	0.9989	-	-	2	0.7126	0.7082	0.8292	0.6568	3.0989	0.5027	2	0.6559	0.6476	0.7866	0.4846	1.4095	0.8705
SPECTF_heart	2	5	0.4278	0.2862	0.4669	0.0643	0.3258	2.1408	2	0.549	0.5224	0.6877	0.2835	0.7426	1.4047	2	0.5677	0.5395	0.7034	0.297	0.758	1.3722	2	0.5279	0.4916	0.6595	0.2565	1.4193	1.497
svmguide1	2	2	0.5367	0.5367	0.732	0.3225	1.1449	1.1835	2	0.5367	0.5367	0.7326	0.3225	1.1449	1.1835	2	0.5571	0.5571	0.7464	0.3318	1.1623	1.1638	2	0.5369	0.5369	0.7327	0.3226	1.1451	1.1832
teaching assistance	3	4	0.5626	0.2164	0.356	0.5742	1.3105	0.26851	3	0.4285	0.2986	0.4967	0.656	2.5724	0.5514	3	0.4986	0.2855	0.4614	0.4846	1.9057	0.8326	5	0.6147	0.1966	0.3322	0.2862	1.2933	1.2051
titanic	2	2	0.6522	0.5581	0.7189	0.6393	2.2277	0.6382	2	0.6522	0.5581	0.7189	0.6393	2.2277	0.6782	3	0.6497	0.5428	0.7045	0.6951	2.1626	0.5529	3	0.5902	0.4279	0.6024	0.6517	1.5011	0.5585
vehicle	4	2	0.5022	0.2576	0.497	0.432	1.4068	0.8958	3	0.5347	0.2528	0.4334	0.4299	1.482	0.8331	5	0.5669	0.2626	0.4409	0.6336	0.3922	0.5757	3	0.5347	0.2528	0.4334	0.4299	1.482	0.8331
breast cancer	2	3	0.5559	0.4132	0.5875	0.2152	0.706	1.8439	2	0.1574	0.5977	2.1946	0.5484	0.4128	0.5862	3	0.6118	0.5107	0.6763	0.6551	0.8035	1.9537	4	0.4709	0.23	0.4055	0.1604	0.7448	1.7706
house vote 84	2	3	0.7509	0.5958	0.7848	0.2059	1.0975	1.6347	2	0.7752	0.6415	0.7818	0.2831	1.0116	1.471	2	0.7583	0.6187	0.7646	0.2801	0.9122	1.4837	2	0.7684	0.6312	0.7742	0.2798	0.9491	1.4867

TABLE VI SUMMARY OF PERFORMANCE INDECES' AND SCORE

Indices		KMA	KMA++	WKMA	H&W KMA
Means	*Rand*	*0.610725*	*0.61878*	*0.60022*	*0.626135*
	Jaccard	*0.418303*	*0.47386*	*0.4902*	*0.47684*
	FM	*0.58125*	*0.635765*	*0.72505*	*0.644905*
	Silhouette	*0.306635*	*0.331235*	*0.37579*	*0.34828*
	Dunn	*0.306635*	*0.331235*	*0.37579*	*0.34828*
	DB	*1.144865*	*1.25812*	*1.123426*	*1/142875*
Rank Score		*9*	*16*	*17*	*18*

H&W K-Means is Hartigan and Wong K-Means

is slightly worse than both Weighted K-Means and Hartigan and Wong K-Means. The K-Means++ algorithm takes into consideration the inter-cluster distances. So while choosing the initial clusters, they are chosen to be farthest from each other so that even after clustering the inter-cluster distance remains high. This may be contributing to both the success and fall of K-Means++ algorithm. For datasets with well-separated clusters, K-Means++ gives better result. On the other hand, for datasets with overlapping clusters or closely located clusters, performance of K-Means++ is poor.

Weighted K-Means algorithm performs best for Rand Index. But with respect to Rank Score, it is second only to Hartigan and Wong K-Means. A close observation of top three indices (external indices) of Table VI shows that the external indices contribute mostly to the Rank Score of Weighted K-Means algorithm. This obviously means that the clustering resulted from Weighted K-Means are closer to the original clustering. However with respect to internal indices are worse than both K-Means++ and Hartigan and Wong algorithm. It is therefore obvious that a close match with the experts' choices may not always result in well separated and compact clusters.

Hartigan and Wong algorithm performs best for two internal and two external indices. The Hartigan and Wong algorithm chooses the initial centroids on the basis of their density and their distances from each other. So it takes care of the density (or compactness) as well as separation of the clusters. Because of this, Hartigan and Wong algorithm gives the best result of the 4 algorithms.

4. CONCLUSION

This work empirically compares four clustering algorithms, K-Means, K-Means++, Weighted K-Means, Hartigan and Wong K-Means, choosing wide variety of real life datasets. The performance indices considered in this paper are both internal and external. It has been seen that all three improvements over K-Means are far better than the original K-Means algorithm. While Weighted K-Means algorithm results in closest approximations to the experts' choices as indicated by the external indices, it may not always result in more compact and well separated clusters. K-Means++ clustering algorithm is good for well separated clusters but not for overlapping and closer clusters. Hartigan and Wong's K-Means algorithm results in slightly better clustering compared to both K-Means++ and Weighted K-Means since it takes into account both density and separation of the clusters. However, all three variations of K-Means studied in this work are either improvements of K-Means with respect to initialization method or assigning weights to the attributes. The K-Means algorithm may be stuck at local optima [1], so may be the three variants of K-Means studied in this work. A separate study [11], with three other different algorithms, showed that the performance of Genetic K-Means algorithm [12] with inverse of Sum of Squared Error (SSE) as the fitness function performed best because it was able to search the global optima. Therefore, a multi-objective Genetic K-Means algorithm considering density (as observed from Hartigan and Wong's algorithm) and maximum inter-cluster distance (as observed from K-Means++ algorithm) along with inverse of SSE could be a better choice to come up with a pareto-optimal solution.

ACKNOWLEDGMENT

This part is suppressed for double blind review.

REFERENCES

[1] C. C. Aggarwal, "An Introduction to Cluster Analysis", In C. C. Aggarwal and C. K. Reddy (Eds.), Data Clustering Algorithms and Applications, pp. 1-28, CRC Press, Boca Raton, FL, 2014.

[2] A. Jain, "Data Clustering: 50 years beyond k-means", Pattern Recognition letters, 31(8):651-666, 2010.

[3] J. B. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability*, Vol. I: Statistics, 281–297, 1967.

[4] D. Arthur and S. Vassilvitskii, K-Means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1927-1035. Society for Industrial and Applied Mathematics, 2007.

[5] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, Automated variable weighting in k-means type clustering, "IEEE Transactions on Pattern Analysis and Machine Intelligence", 27(5):657-668, 2005.

[6] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm", Journal of the Royal Statistical Society, Series C (Applied Statistics), 28(1), 100-108, 1979.

[7] C. D. Manning, P. Raghavan, and H. Schutze, "Introduction to Information Retrieval", Volume 1, Cambridge University Press, Cambridge, 2008.

[8] C. L. Blake, and J. C. Merz, UCI repository of machine learning databases (machine-readable data repository), Department of Information and Computer Science, University of California, Irvine, 1999.

[9] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "Cluster validity methods: part I", SIGMOD Rec., Vol. 31, No. 2, pp. 40-45, 2002.

[10] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "Cluster validity methods: part II", SIGMOD Rec., Vol. 31, No. 3, 19-27, 2002.

[11] This reference is suppressed for double blind review.

[12] K. Krishna and M. N. Murty, "Genetic k-means algorithm". IEEE Transactions on Systems, Man, Cybernetics, Part(B): Cybernetics, 29(3):433-439, 1999.