Journal of Water Resource and Protection, **2012, 4, 560-566**
http://dx.doi.org/10.4236/jwarp.2012.47065 Published Online July 2012 (http://www.SciRP.org/journal/jwarp)

Regionalization of River Basins Using Cluster Ensemble

Sangeeta Ahuja
Division of Computer Applications, IASRI (ICAR), New Delhi, India
Email: reach2san@yahoo.com, sangeeta@iasri.res.in

Received March 17, 2012; revised April 23, 2012; accepted May 26, 2012

ABSTRACT

In the wake of global water scarcity, forecasting of water quantity and quality, regionalization of river basins has attracted serious attention of the hydrology researchers. It has become an important area of research to enhance the quality of prediction of yield in river basins. In this paper, we analyzed the data of Godavari basin, and regionalize it using a cluster ensemble method. Cluster Ensemble methods are commonly used to enhance the quality of clustering by combining multiple clustering schemes to produce a more robust scheme delivering similar homogeneous basins. The goal is to identify, analyse and describe hydrologically similar catchments using cluster analysis. Clustering has been done using RCDA cluster ensemble algorithm, which is based on discriminant analysis. The algorithm takes H base clustering schemes each with K clusters, obtained by any clustering method, as input and constructs discriminant function for each one of them. Subsequently, all the data tuples are predicted using H discriminant functions for cluster membership. Tuples with consistent predictions are assigned to the clusters, while tuples with inconsistent predictions are analyzed further and either assigned to clusters or declared as noise. Clustering results of RCDA algorithm have been compared with Best of k-means and Clue cluster ensemble of R software using traditional clustering quality measures. Further, domain knowledge based comparison has also been performed. All the results are encouraging and indicate better regionalization of the Godavari basin data.

Keywords: K-Means; Cluster Ensemble; Hydrology; Runoff; Cultivation Area; Precipitation; Field Capacity

1. Introduction

Estimating design flow of ungauged basins is very crucial in the planning and management of hydraulic and water resources engineering. Regionalization for identifying homogeneous hydrologic regions is a well-accepted technique in this area. Regionalization is defined as determination of hydrologically similar units, and is one of the most challenging tasks in surface hydrology. In recent years several new mathematical and computational tools have been explored for this task [1].

Regionalization is done for estimating design flow in ungauged basins which is frequently encountered in the design and planning of hydraulic and water resources engineering [2]. The hydrologic regionalization technique is to infer required data in ungauged catchments from neighbour catchments where hydrologic data have been collected (e.g. Nathan and McMahon, 1990; Bullock and Andrews, 1997; Hall and Minns, 1999).

Runoff predictions in ungauged catchments are determined by regionalization. Development of practical runoff prediction methods are important for assessing water resources in an ungauged or poorly gauged catchment which is usually located in headwater regions [2]. Excess runoff can lead to flooding, which occurs when there is too much precipitation.

Catchment shows a wide range of response behaviour, therefore, Regionalization is utilized for searching the hydrological similarity of catchments to characterize each catchment [3].

2. Background and Related Work

We present the related work with respect to two aspects *i.e.* the techniques used for regionalization in hydrology studies and the techniques of cluster ensemble. Subsequently we describe the discriminant based cluster ensemble algorithm (RCDA) used in this work.

2.1. Regionalization

Hydrological similarity of catchments is identified and analyzed in the paper [3] by using the concept of Self-Organizing Maps (SOM). SOM are plotted by utilizing the hierarchical clustering algorithm of cluster analysis.

A regional formula has been developed by the authors, using gauged flows and basin topographic characteristics in order to estimate the design flows in ungauged areas within the homogeneous region [1].

Principal component and cluster analyses were used to delineate into homogeneous regions. Statistical tests demonstrate that the design flows are significantly related

to the topographic variables at 5% significance level and the delineation of homogeneous regions can enhance the performance of regional formulae to estimate design flow.

Different regionalization methods were investigated in the paper [2], for modeling daily runoff in ungauged catchments for selecting donor catchments whose entire set of parameter values are used for target ungauged catchment by determine the spatial proximity, physical similarity and integrated similarity.

Regionalization of runoff formation by aggregation of hydrological response units for the representative elementary areas (REAs), which are defined as homogeneous, the hydrologically effective parameters can be clearly assigned. Aggregation approaches were used to analyse the research regions which differ in the composition of their natural attributes. The purpose of the regional comparison is to reveal to what extent it is possible to apply the regionalization strategy independent of the region and scale [4].

These analyses substantiate the fact that it is possible to achieve plausible results with the regionalization approaches which have been developed, provided that geoinformation for the entire region is available. The comparison shows that the regionalization approaches are independent of area and scale and these regionalization procedures significantly improve the quality of simulation of the water balance for large drainage basins, with a significant reduction of the relational-geometric configurations [4].

2.2. Cluster Ensemble Approach

Motivation of Cluster Ensemble technique arises because of different clustering schemes that are obtained by application of different clustering algorithms, or by varying the parameters of the same clustering algorithm. For example, in k-means algorithm, which is one of the most used clustering algorithms, variations in results arise because of the inherent randomization. Further, each algorithm performs differently depending upon the biases and assumptions associated with it.

Under such circumstances, it is very difficult to ascertain suitability of an algorithm for an application. Cluster ensemble techniques aims to improve the clustering scheme by intelligently combining multiple schemes. This technique has caught attention of researchers in computer science community as it has found to substantially improve the robustness, stability, accuracy and quality of resulting clustering scheme [5-9]. An informative survey of various cluster ensemble techniques can be found in [5]. The problem of cluster ensemble is formally defined below.

Let D denote a data set of N, d-dimensional vectors $Xi = \left\langle X_i^1, X_i^2, \cdots X_i^d \right\rangle$ where $i = 1, N$, each representing an object. D is subjected to a clustering algorithm which delivers a partition (*i.e.* a clustering scheme) π' consisting of K clusters, *i.e.* ($\pi' = C_1, C_2, \ldots, C_K$). Let λ' be the function of π; ($\lambda' : \rightarrow \{1,K\}$) that yields labeling for each of the N objects in D. Let $\{\pi_1', \pi_2', \cdots, \pi_H'\}$ be H partitions of D obtained by applying either same clustering algorithm on D or by applying H different clustering algorithms.

Before combining the schemes, it is necessary to establish the correspondence between the clusters of different schemes and relabel the corresponding clusters. Let $\{\lambda_1, \lambda_2, \cdots, \lambda_H\}$ be the set of corresponding labeling of H clustering schemes on D. The problem of cluster ensemble is to derive a consensus function Γ, which combines H partitions and delivers a clustering π_f with a promise that π_f is more robust than any of constituent H partitions and *best* captures the natural structures in D. **Figure 1** shows the process of construction of cluster ensemble.

It is the design of Γ that distinguishes different cluster ensemble algorithms to a large extent. Hyper graph partitioning [5] voting approach [10], mutual information [5, 11], co-associations [12-14] are some of the well-established approaches for building consensus functions.

2.3. RCDA (Robust Clustering Using Discriminant Analysis)

RCDA [15] is a recent algorithm for generating a robust clustering scheme using discriminant analysis. Robust Clustering Using Discriminant Analysis (*RCDA*) algorithm takes H partitions as input with K clusters in each partition and delivers a robust partition with same number of clusters, and noise, if any. It operates in three phases. In the first phase clusters in each partition are relabeled to establish correspondence in H partitions. In the second phase the algorithm constructs a discriminant function for each partition, thereby resulting in H discriminant functions. Cluster label of each tuple in dataset D is predicted by each of the H discriminant functions

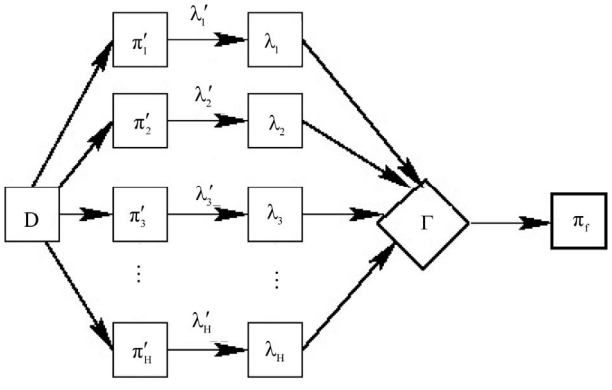

Figure 1. The process of cluster ensemble.

resulting in N X H label matrix (L). This is a compute intensive phase of the algorithm and needs no user parameter. Finally, in the third phase tuples with consistent labels are assigned to clusters in the final partition. Tuples with low consistency are refined and the leftover tuples are reported as noise. Different phases of RCDA algorithm is shown pictorially in **Figure 2**.

3. Regionalization Using RCDA

In this study the hydrological similarity of a catchment area has been investigated with respect to their response behaviour by using RCDA Algorithm. The goal is to identify, analyse and describe hydrologically similar catchments/regions by using the catchment characteristics such as Elevation, Precipitation, Aridity Index, Slope, Field Capacity and Stream Density.

Data from Godavari basin is processed using RCDA algorithm in order to regionalize the river basin. The data consists of 331 tuples and six attributes viz., Elevation, Precipitation, Aridity Index, Slope, Field Capacity and Stream Density respectively.

Since the numbers of regions are not known, the data is pre-processed using domain knowledge to estimate the number of clusters. Intuitively, the regions with same runoff/Catchment Area ratio should fall in same cluster. Based on this idea, runoff/Catchment Area ratio was computed for all tuples. The mnemonics for the bins for bin-widths (0.08 and 0.04) are represented in **Table 1** and **Table 2** respectively. The frequency charts for two bin-widths (0.08 and 0.04) respectively were constructed as shown in **Figure 3** and **Figure 4**.

It can be seen from the **Figure 3** that last three bins (numbered 6, 7 and 8 along x-axis) consists of only 0,1 and 2 tuples respectively. So, we eliminated the three noisy tuples and also it can be seen from the **Figure 4** that last six bins (numbered (11, 12, 13, 15), 14, 16) consists of only 0, 1, 2 and 3 tuples respectively. So, we eliminated the six noisy tuples.

This analysis indicates that either there are five or nine regions in the Godavari basin. We applied RCDA algorithm to cluster 328 tuples after removing three noisy tuples in case of five regions and cluster 325 tuples after removing six noisy tuples in case of nine regions.

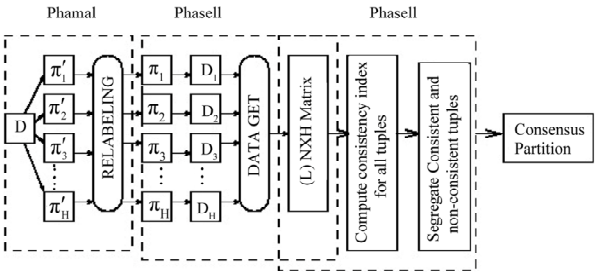

Figure 2. Three phases of RCDA algorithm.

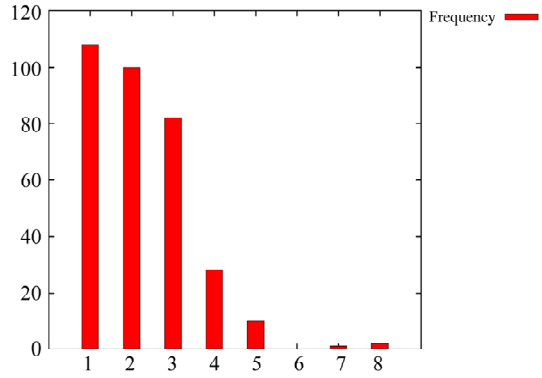

Figure 3. Frequency Chart for runoff/Catchment Area ratio with bin width of 0.08.

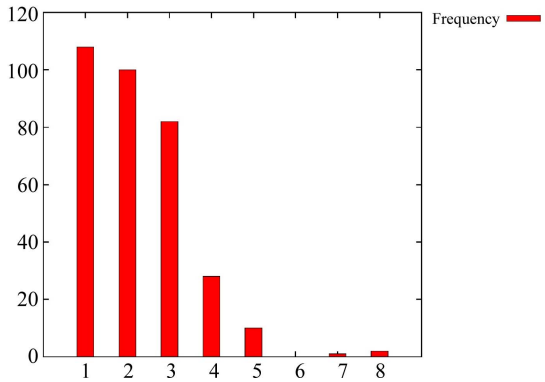

Figure 4. Frequency Chart for runoff/Catchment Area ratio with bin width of 0.04.

Table 1. Description of data sets—mnemonics used for Bins.

Bins	Mnemonic
0.03 - 0.11	1
0.11 - 0.19	2
0.19 - 0.27	3
0.27 - 0.35	4
0.35 - 0.43	5
0.43 - 0.51	6
0.51 - 0.59	7
0.59 - 0.67	8

4. Experimental Section

RCDA (Robust Clustering Using Discriminant Analysis) algorithm was implemented in Windows environment as multi-threaded C++ program. R package (V 2.13.0) was used for statistical functions. Dual core Intel(R) machine (2.20 GHz, 4 GB RAM) was used for executing programs. In this section we describe the goals and methodology of experiments.

Having determined two possibilities for the number of clusters in Godavari basin data, we applied RCDA algo-

Table 2. Description of data sets—mnemonics used for Bins.

Bins	Mnemonic
0.03 - 0.07	1
0.07 - 0.11	2
0.11 - 0.15	3
0.15 - 0.19	4
0.19 - 0.23	5
0.23 - 0.27	6
0.27 - 0.31	7
0.31 - 0.35	8
0.35 - 0.39	9
0.39 - 0.43	10
0.43 - 0.47	11
0.47 - 0.51	12
0.51 - 0.55	13
0.55 - 0.59	14
0.59 - 0.63	15
0.63 - 0.67	16

rithm to get the clustering schemes. We describe the results in two sections for the two possibilities.

Validation of results is performed both at computational and domain level. Computational validation of results is performed by comparing the SSE (Sum of Squared Error) of the clustering scheme obtained by RCDA, with another cluster ensemble method available in R software and the best of the constituent clustering scheme. The scheme with the lowest SSE is the best clustering scheme (optimum partition). The domain level validation is performed by comparing the purity and NMI of the obtained scheme with the frequency distribution shown in **Figure 3** and **Figure 5**, which is taken as gold standard. In subsequent sections, we detail the computation of SSE and Purity.

4.1. Computing SSE

For measuring the quality of clustering, we use the Sum of Squared error (SSE), which is also known as scatter. In other words, we calculate the error of each data point, *i.e*. its eucleidean distance to the closest centroid, and then compute the total sum of squared errors. If we have two different sets of clusters by two different algorithms (schemes), we prefer the one with the smallest squared error. The SSE [16] is formally defined as

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} dist\left(\bar{x}_i - x\right)^2 \qquad (1)$$

where *dist* is the standard Euclidean distance between the two objects in eucleidean space, C_i = i^{th} cluster, x is a point in C_i and $\bar{x}_i$ is the mean (centroid) of the i^{th} cluster[1].

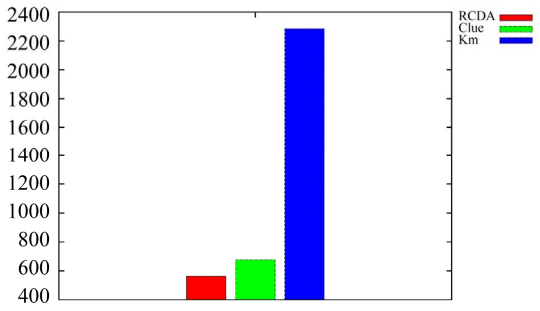

Figure 5. Comparison of SSE of RCDA, Clue and Best of K-means (Km) algorithm for *K* = 5.

4.2. Computing Purity

For each cluster, the class distribution of the data is calculated first, *i.e*. for cluster j we compute p_{ij}, the probability that a member of cluster i belongs to belong j as $p_{ij} = m_{ij}/m_i$, where m_i is the number of objects in cluster i and m_{ij} is the number of objects of class j in cluster i.

The purity of cluster i is defined in [16] as

$$p_i = \max_j \left(p_{ij} \right) \qquad (2)$$

The overall purity of a partition is

$$\mathrm{Purity} = \sum_{i=1}^{K} \frac{m_i}{m} p_i \qquad (3)$$

In general, larger value of purity indicates better quality of the solution.

4.3. Computing NMI (Normalized Mutual Information)

Intuitively, the optimal combined clustering should share the most information with the original clusterings. Thus NMI has been used by researchers to measure cluster quality [11].

Let A and B be the random variables described by the cluster labeling $\lambda(a)$ and $\lambda(b)$ with $k(a)$ and $k(b)$ groups respectively. Let I(A,B) denote the mutual information between A and B, H(A), H(B) denote the entropy of A and B respectively. Then normalized mutual information (NMI) is defined as follows

$$\mathrm{NMI(A,B)} = 2\ \mathrm{I(A,B)}/(\mathrm{H(A)} + \mathrm{H(B)}) \qquad (4)$$

Clearly, the value lies between [0, 1] and NMI(A,A) = 1.

Equation (4) is estimated by the sampled entities provided by the clustering. Let $n^{(h)}$ be the number of objects in cluster c_h according to $\lambda(a)$ and let n_g be the number of objects in cluster c_g according to $\lambda(b)$. Let n_h^g be the

[1]Here, in our case x is the tuple consists of six attributes (catchment characteristics) viz., Elevation, Precipitation, Aridity Index, Slope, Field Capacity and Stream Density and $\bar{x}_i$ is the centroid of the i^{th} cluster.

number of objects in cluster c_h according to $\lambda(a)$ as well as in cluster c_g according to $\lambda(b)$. The normalized mutual information criteria φ(NMI) is computed as follows

$$\varphi^{NMI}_{(\lambda(a),\lambda(b))} = \frac{2}{n}\left(\sum_{h=1}^{k(a)}\sum_{g=1}^{k(b)}\left(n_g^h\right)\log_{k(a)k(b)}\frac{n_g^{(h)}\cdot n}{n^h\cdot n_g}\right) \quad (5)$$

In our context, $k(a) = k(b) = k$.

4.4. Results with 5 Clusters

We experimented the dataset with RCDA cluster ensemble algorithm [15] for K (number of clusters = 5) with varying the number of partitions (H = 2, 4, 6, 8, 10, 12, 14, 16 and 18) respectively. Here, we get the optimum partition H = 8, because at this value of partition, we obtained the lowest value of **SSE** (Sum of Squared Error) and maximum (improved) clustering quality. The comparison of the RCDA algorithm with Best of K-means (Km) and Clue Ensemble obtained from R software [17] have been done by determining the centroids as shown in **Table 3** and Total SSE (Sum of Squared Error) of each algorithm as shown in **Figure 5**. Moreover, comparisons of RCDA algorithm with Best of K-means (Km) and Clue Ensemble obtained from R software [17] have been done in terms of measuring Purity and NMI (Normalized Mutual Information) as shown in **Figure 6**.

Table 3 shows the centroids of each cluster of RCDA and Clue algorithm for K = 5 number of clusters. ELV, PPT, AI, Sl, FC and SD in **Table 3** represents the Elevation, Precipitation, Aridity Index, Slope, Field Capacity and Stream Density respectively.

4.5. Results with 9 Clusters

Similarly, we experimented the dataset with RCDA cluster ensemble algorithm [15] for K (number of clusters = 9) with varying the number of partitions (H = 2, 4, 6, 8, 10, 12, 14, 16 and 18) respectively. Here, we get the optimum partition H = 8, because at this value of partition, we obtained the lowest value of SSE (Sum of Squared Error) and maximum (improved) clustering quality.

The comparison of the RCDA algorithm with Best of K-means (Km) and Clue Ensemble obtained from R software [17] have been done by determining the centroids as shown in **Table 4** and total SSE of the each algorithm as shown in **Figure 7**. Moreover, comparisons of RCDA algorithm with Best of K-means (Km) and Clue Ensemble obtained from R software [17] have been done in terms of measuring Purity and NMI (Normalized Mutual Information) as shown in **Figure 8**.

5. Discussion of Results

We observed from the **Figure 5** and **Figure 7** that total SSE of RCDA algorithm is less than as compared to the total SSE of Clue Algorithm which clearly describes that variability reduces in case of RCDA algorithm as compared to clue algorithm which is the characteristics of good quality clustering.

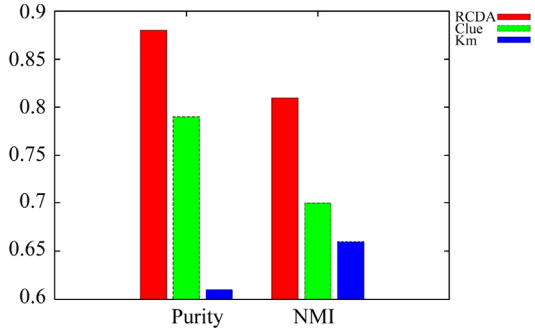

Figure 6. Comparison of purity and NMI of RCDA, Clue and Best of K-means (Km) algorithm for K = 5. \label {Fig_PN5}.

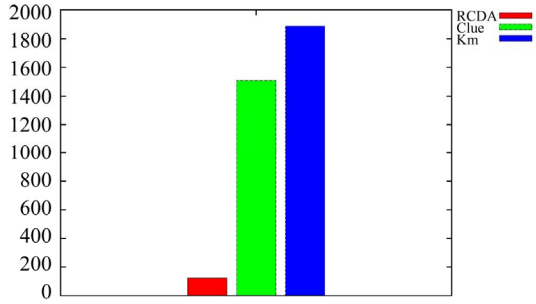

Figure 7. Comparison of SSE of RCDA and Clue algorithm for K = 9.

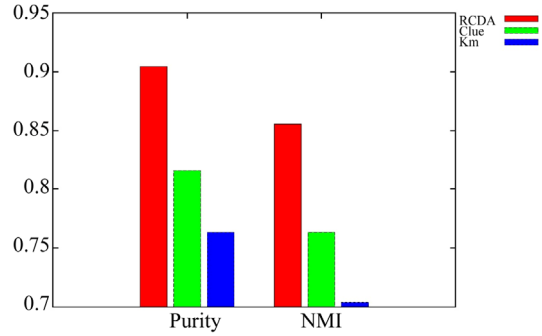

Figure 8. Comparison of purity and NMI of RCDA, Clue and Best of K-means (Km) algorithm for K = 9.

Moreover, the Purity and NMI of RCDA algorithm improves as compared to Best of K-means (Km) and Clue cluster Ensemble obtained from R software as shown in **Figure 6** and **Figure 8**.

Finally, from the above it is concluded that K = 9 with H = 8 and 325 number of tuples is the optimum case. Since, SSE for both the cases K = 5 and K = 9 of RCDA algorithm is less as compared to Clue and Best of K-means (Km) algorithm, but the SSE is very less in K = as shown

Table 3. Centroids of each cluster for RCDA, Clue and Best of K-means (Km) algorithm for $K = 5$.

Algorithm	ELV	PPT	AI
	Cluster 1		
RCDA	454.686098	1009.700488	2.565610
Clue	368.907349	992.352289	2.683614
Km	396.710660	1051.251892	2.455225
	Cluster 2		
RCDA	253.862644	1250.008851	2.045402
Clue	213.641786	1337.895000	1.872857
Km	396.643784	200.416216	2.243243
	Cluster 3		
RCDA	499.338548	1317.570484	1.987419
Clue	583.356491	1384.648246	1.699649
Km	474.658000	1046.805333	2.417000
	Cluster 4		
RCDA	415.963750	895.337083	2.923333
Clue	49.666623	753.095065	3.082857
Km	369.956111	1303.237222	1.856111
	Cluster 5		
RCDA	385.400625	1010.806250	2.802187
Clue	264.685556	1363.166667	2.305556
Km	386.476792	1170.911321	2.347453

Table 4. Centroids of each cluster for RCDA, Clue and Best of K-means (Km) algorithm for $K = 9$.

Algorithm	ELV	PPT	AI
	Cluster 1		
RCDA	304.728148	1251.386667	2.077407
Clue	568.021707	1367.127073	1.738049
Km	528.729299	801.104068	2.666991
	Cluster 2		
RCDA	551.496296	834.513889	2.880556
Clue	597.165897	825.544615	2.799231
Km	324.264769	1265.239846	2.049538
	Cluster 3		
RCDA	316.341481	1000.493333	2.691111
Clue	274.213509	1093.276491	2.508772
Km	340.568000	1360.779143	1.784286
	Cluster 4		
RCDA	462.077500	922.229583	2.862708
Clue	253.567727	1398.051591	1.763409
Km	450.545429	808.076286	3.167714
	Cluster 5		
RCDA	236.166197	1297.978310	1.970845
Clue	503.114103	686.310256	3.327949
Km	420.485789	782.421053	3.460000
	Cluster 6		
RCDA	565.201000	814.138000	2.709000
Clue	711.091818	1375.290000	1.617273
Km	295.265294	1331.299412	1.807647
	Cluster 7		
RCDA	759.509375	1237.728125	1.816250
Clue	181.243784	1262.656486	1.998649
Km	481.933000	940.910500	2.678000
	Cluster 8		
RCDA	473.920000	1597.883810	1.528571
Clue	391.232727	1784.505455	1.730909
Km	671.832353	1312.657647	1.675294
	Cluster 9		
RCDA	382.813333	1079.631667	2.767500
Clue	409.789348	938.381739	2.843696
Km	223.790179	1234.093929	2.076786

Continued

Algorithm	Sl	FC	SD
	Cluster 1		
RCDA	0.019019 15	0.370370	0.103417
Clue	0.031683 17	7.814634	0.071464
Km	0.016468 14	5.872376	0.103357
	Cluster 2		
RCDA	0.015907	152.312963	0.110959
Clue	0.018077	189.923077	0.095857
Km	0.021815	151.692308	0.102434
	Cluster 3		
RCDA	0.013370	261.585185	0.024368
Clue	0.017737	179.547368	0.056450
Km	0.026857	156.571429	0.096325
	Cluster 4		
RCDA	0.013375	155.39375	0 0.014827
Clue	0.024159	154.565909	0.045808
Km	0.013571	147.714286	0.013118
	Cluster 5		
RCDA	0.018718	144.225352	0.009158
Clue	0.012051	150.769231	0.071786
Km	0.015737	142.105263	0.014265
	Cluster 6		
RCDA	0.019100	312.000000	0.174771
Clue	0.046545	177.21818	0.095623
Km	0.035353	174.705882	0.010749
	Cluster 7		
RCDA	0.046688	178.993750	0.107478
Clue	0.018703	145.945946	0.033421
Km	0.018950	283.905000	0.142880
	Cluster 8		
RCDA	0.058810	160.952381	0.061680
Clue	0.028091	154.545455	0.062202
Km	0.067765	183.182353	0.083542
	Cluster 9		
RCDA	0.015667	236.850000	0.102557
Clue	0.015348	206.369565	0.068397
Km	0.014125	135.000000	0.008019

in **Figure 5** and **Figure 7**. Similarly, the Purity and NMI of RCDA algorithm for both the cases $K = 5$ and $K = 9$ is more as compared to Clue and Best of K-means (Km) algorithm, but it improves much more in case of $K = 9$ which indicates that more homogeneous catchments are clustered using RCDA algorithm with $K = 9$ number of clusters.

6. Acknowledgements

Expressing my sincere thanks to Dr. Vasudha Bhatnagar, Head, University of Delhi, Delhi, India and Dr. Subhash Chander, Professor (Retd.) of Water Resources, Civil Engineering Department, IIT Delhi, India for their help and encouragement for the production of this manuscript.

REFERENCES

[1] P.-S. Yu, H.-P. Tsai, S.-T. Chen and Y.-C. Wang, "Estimation of Design Flow in Ungauged Basins by Regionalization," Department of Hydraulic and Ocean Engineering, National Cheng Kung University, Taiwan, 2005.

[2] Y. Zhang and F. Chiew, "Evaluation of Regionalization Methods for Predicting Runoff in Ungauged Catchments in Southeast Australia," CSIRO Water for a Healthy Country National Research Flagship, CSIRO Land and Water 13-1, 2009.

[3] R. Ley, M. C. Casper, H. Hellebrand and R. Merz, "Catchment Classification by Runoff Behaviour with Self-Organizing Maps (SOM)," *Journal of Hydrology and Earth System Sciences*, Vol. 15, 2011, pp. 2947-2962. doi:10.5194/hess-15-2947-2011

[4] G. Busch, J. Sutmoller and G. Gerold, "Regionalization of Runoff Information by Aggregation of Hydrological Response Units: A Regional Comparison," *Proceedings of a Conference Regionalization in Hydrology*, Vol. 254, 1997.

[5] R. Ghaemi, N. Sulaiman, H. Irahim and N. Mustapha, "A Survey: Clustering Ensembles Techniques," *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 38, No. 2, 2002, pp. 2070-3740.

[6] X. Hu and I. Yoo, "Cluster Ensemble and Its Applications in Gene Expression Analysis," *Proceedings of Second Asia-Pacific Bioinformatics Conference*, Vol. 29, 2004, pp. 297-302.

[7] A. Topchy, B. Minaei-Bidgoli, A. K. Jain and W. F. Punch, "Adaptive Clustering Ensembles," *Proceedings of the* 17*th International Conference on Pattern Recognition*, Vol. 1, 2004, pp. 272-275. doi:10.1109/ICPR.2004.1334105

[8] A. Topchy, A. K. Jain and W. Punch, "A Mixture Model for Clustering Ensembles," *Proceedings SIAM Conference on Data Mining*, 2004, pp. 379-390.

[9] M. D. Frossyniotis and A. Stafylopatis, "A Multi-Clustering Fusion Algorithm," *SETN*'02 *Proceedings of the Second Hellenic Conference on AI*: *Methods and Applications of Artificial Intelligence*, Springer, London, 2002.

[10] B. Fischer and J. M. Buhmann, "Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, 2003, pp. 513-518. doi:10.1109/tpami.2003.1190577

[11] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, Vol. 3, 2002, pp. 583-617.

[12] A. L. N. Fred, "Finding Consistent Cluster in Data Partitions," *Proceedings of* 2*nd International Workshop on Multiple Classifier Systems*, Vol. 2096, 2001, pp. 309-318.

[13] A. L. N. Fred and A. K. Jain, "Data Clustering Using Evidence Accumulation," *Proceedings of International Conference on Pattern Recognition*, Vol. 4, 2002, pp. 276-280.

[14] A. Topchy, A. K. Jain and W. Punch, "Combining Multiple Weak Clusterings," *Proceedings of the* 3*rd IEEE International Conference on Data Mining*, 19-22 November 2003, pp. 331-338. doi:10.1109/ICDM.2003.1250937

[15] V. Bhatnagar and S. Ahuja, "Robust Clustering Using Discriminant Analysis," *Proceedings of International Industrial Conference on Data Mining*, Vol. 6171, 2010, pp. 143-157.

[16] P. N. Tan, V. Kumar and M. Steinbach, "Introduction to Data Mining," Pearson, March 2006.

[17] http://cran.r-project.org/web/packages/clue/clue.pdf

