Weight-Improved K-Means-Based Consensus Clustering

Yanhua Wang, Laisheng Xiang, and Xiyu Liu(✉)

School of Management Science and Engineering, Shandong Normal University, Jinan, China
15554130027@163.com, xls3366@163.com, sdxyliu@163.com

Abstract. Many consensus clustering methods ensemble all the basic partitionings (BPs) with the same weight and without considering their contribution to consensus result. We use the Normalized Mutual Information (NMI) theory to design weight for BPs that participate in the integration, which highlights the contribution of the most diverse BPs. Then an efficient approach K-means is used for consensus clustering, which effectively improves the efficiency of combinatorics learning. Experiment on UCI dataset iris demonstrates the effective of the proposed algorithm in terms of clustering quality.

Keywords: Consensus clustering · K-means · Basic partitionings

1 Introduction

There is no single clustering algorithm can performs best for all data sets [1], and can discover all types of cluster shapes and structures [2]. Consensus clustering approached are able to integrate multiple clustering solutions obtained from different data sources into a unified solution, and provide a more robust, stable and accurate final result [3]. However, the previous research still has some limitations.

Firstly, high quality BPs are beneficial to the performance of consensus clustering yet the partitions with poor quality will lead to worse consensus result. But most studies tend to integrate all BPs, and they do not filter poor BPs. Secondly, the diversity between BPs also have great impact on consensus clustering, diverse BPs which means the BP that has different mutual information with other BPs will have different contribution to the consensus result. However, there are few references explore impact of the number of BPs to consensus clustering neither did they take into account the diversity of BPs into the integration process.

We proposed weight-improved K-means-based consensus clustering (WIKCC). Firstly, we design weight for each BP participating in the integration. Specifically, we generate multiple BPs and measure the quality of each BP using normalized Rand index R_n [6], and sort the BPs in the increasing order of R_n, then we explore the influence of the number of BPs on consensus clustering, based on the above exploration we can choose an appropriate number of better BPs for consensus clustering, which can minimize the number of BPs in quality assurance. After that we construct the co-occurrence matrix with the selected BPs, and calculate the similarity of two

Q. Zu and B. Hu (Eds.): HCC 2017, LNCS 10745, pp. 46–52, 2018.
https://doi.org/10.1007/978-3-319-74521-3_6

BPs with Normalized Mutual Information (NMI) method [4] according to the co-occurrence matrix. Then weight of each BP is designed according to NMI values which reflect a single BP to overall BPs' similarity. K-means-based method [2] make special attention for its simplicity and high efficiency. So we transform consensus clustering to K-means clustering.

2 Weight Design Based on the Normalized Mutual Information

Mutual information is used to measure the shared information of the two distributions. We compute the NMI between two BPs, and the greater the value of NMI means the lower difference, which will result in lower w_i.

Given two BPs results π_i with K_i clusters, $\pi_i = \left\{ C_1^{(i)}, C_2^{(i)}, \ldots, C_{K_i}^{(i)} \right\}$ and π_j with K_j clusters, $\pi_j = \left\{ C_1^{(j)}, C_2^{(j)}, \ldots C_{K_j}^{(j)} \right\}$ the mutual information between two results is defined as follows:

$$\mathrm{NMI}(\pi_i, \pi_j) = \frac{2I_1(\pi_i, \pi_j)}{I_2(\pi_i) + I_2(\pi_j)} \tag{1}$$

$$I_1(\pi_i, \pi_j) = \sum_h \sum_l \frac{\left| C_h^{(i)} \cap C_l^{(j)} \right|}{n} \log \frac{n \left| C_h^{(i)} \cap C_l^{(j)} \right|}{\left| C_h^{(i)} \right| |C_l^{(j)}|} \tag{2}$$

$$I_2(\pi_i) = -\sum_h \frac{|C_h^{(i)}|}{n} \log \frac{|C_h^{(i)}|}{n} \tag{3}$$

$$I_2(\pi_j) = -\sum_l \frac{|C_l^{(j)}|}{n} \log \frac{|C_l^{(j)}|}{n} \tag{4}$$

For a single BP the average mutual information can be defined as:

$$\mathrm{H}(\pi_i) = \frac{1}{r-1} \sum_{k=1, k \neq i}^{r} NMI(\pi_i, \pi_k), (i = 1, 2, \ldots r) \tag{5}$$

Where $h \in \{1, 2, \ldots, k_i\}, 1 \in \{1, 2, \ldots, k_j\}$ is one of the cluster result label of π_i and π_j, $\left| C_h^{(i)} \right|, \left| C_l^{(j)} \right|$ respectively represent the number of the data set belong to cluster $C_h^{(i)}$ in π_i and $C_l^{(j)}$ in π_j, $\left| C_h^{(i)} \cap C_l^{(j)} \right|$ is the number of the dataset belong both to $C_h^{(i)}$ and $C_l^{(j)}$, r is the number of the BPs.

The greater $H(\pi_i)$ indicate that cluster member π_i share more information with other cluster members. The weight is defined as:

$$w_m' = \frac{1}{H(\pi_i)} \tag{6}$$

The normalized form is defined as:

$$w_m = \frac{w'_m}{\sum_{m=1}^{r} w'_m} \tag{7}$$

The weight is bigger as the greater diversity between two base clustering.

3 The Weight-Improved K-Means-Based Consensus Clustering

In this section, we first introduce the co-occurrence matrix which is used for records the situation of sharing dataset between two BPs. Table 1 shows an example.

Table 1. The co-occurrence matrix

π^* \ π_i	$C_1^{(i)}$	$C_2^{(i)}$	…	$C_{K_i}^{(i)}$	Σ
C_1	$n_{11}^{(i)}$	$n_{12}^{(i)}$	…	$n_{1K_i}^{(i)}$	n_{1+}
C_2	$n_{21}^{(i)}$	$n_{22}^{(i)}$	…	$n_{2K_i}^{(i)}$	n_{2+}
	.	.	…	.	.
C_K	$n_{K1}^{(i)}$	$n_{K2}^{(i)}$	…	$n_{KK_i}^{(i)}$	n_{K+}
Σ	$n_{+1}^{(i)}$	$n_{+2}^{(i)}$	…	$n_{+k_i}^{(i)}$	n

In Table 1, BPs: π^* and π_i contain k and k_i clusters respectively, $n_{KK_i}^{(i)}$ represents the number of the objects that belongs to both C_K and $C_{K_i}^{(i)}$, then let $n_{k_+} = \sum_{j=1}^{K_i} n_{kj}^{(i)}, 1 \le j \le K_i, 1 \le k \le K, P_{kj}^{(i)} = n_{kj}^{(i)}/n, p_{k+} = n_{k+}/n, and\, p_{+j}^{(i)} = n_{+j}^{(i)}/n.$ We can obtain a normalized co-occurrence matrix (NCM), based on which we can compute the centroid of the K-means clustering.

K-means algorithm cannot directly run on the co-occurrence matrix, so a binary data set is introduced to represent the result of r BPs. The binary data set $X_l^{(b)} = \{x_l^{(b)} | 1 \le l \le n\}$ as follows:

$$x_l^{(b)} = \left\langle x_{l,1}^{(b)}, \ldots, x_{l,i}^{(b)}, \ldots, x_{l,r}^{(b)} \right\rangle, \text{ with} \tag{8}$$

$$x_{l,i}^{(b)} = \left\langle x_{l,i1}^{(b)}, \ldots, x_{l,ij}^{(b)}, \ldots x_{l,iKi}^{(b)} \right\rangle, \text{ and} \tag{9}$$

$$x_{l,ij}^{(b)} = \begin{cases} 1, \textit{ if object } l \textit{ belongs to the cluster } C_j \textit{ in } \pi_i \\ 0, \textit{ otherwise} \end{cases}, \tag{10}$$

Where $x_l^{(b)}$ is an $n \times \sum_{i=1}^{r} K_i$ binary data set matrix with $\left| x_{l,i}^{(b)} \right| = 1$.

We use the K-means algorithm to integrate the BPs, suppose r BPs are integrated to a result π^*, m_k represent the centroid of the C_k *in* π^* as follows:

$$m_k = \langle m_{k,1}, \ldots, m_{k,i}, \ldots, m_{k,r} \rangle, with \tag{11}$$

$$m_{\mathrm{k,i}} = \langle m_{k,i1}, \ldots, m_{k,ij}, \ldots, m_{k,iK_i} \rangle, \tag{12}$$

The centroids of the K-means on X^b are represented as follows:

$$m_{k,i} = \left\langle \frac{p_{k1}^{(i)}}{p_{k+}}, \ldots, \frac{p_{kj}^{(i)}}{p_{k+}}, \ldots, \frac{p_{kK_i}^{(i)}}{p_{k+}} \right\rangle, \forall k, i. \tag{13}$$

The centroids can be computed by the Co- occurrence matrix, and m_k is a vector of $\sum_{i=1}^{r} k_i$ dimension. The element in the vector is computed by the number of shared data set between current cluster and all of the clusters of BPs.

By using the co-occurrence matrix and the binary data set the consensus clustering are transformed to the K-means clustering, that is:

$$\max \sum_{i=1}^{r} w_i U(\pi, \pi^*) = min \sum_{k=1}^{k} \sum_{xl \in C_k} f\left(x_l^{(b)}, m_k\right) \tag{14}$$

As shown in Fig. 1. In BPs generation phase, classic partition clustering method K-means is used, different initial number of cluster k, to generate diversified BPs. In consensus clustering phase, after generating the BPs and computing the weight for each clustering member, we can obtain the weighted co-occurrence matrix, and then we can

Algorithm WIKCC

Require:

Input: a data set of known class label

Ensure:

1. Using K-means to generate r BPs called $\prod$
2. Construct the binary data set $X^{(t)}$ from $\prod$
3. The Co- occurrence matrix is constructed by using BPs
4. Compute the weight of r BPs as $\{w_1, w_2, ..., w_r\}$ by using NMI
5. Reconstruct the binary data set $X^{(t)'}$ using weight
6. Run K-means to cluster $X^{(t)'}$ into K clusters and get π^*
7: Return π^*

Fig. 1. Algorithm WIKCC

get the weighted binary dataset $X^{(b)\prime}$, by running K-means on weighted binary dataset $X^{(b)\prime}$, we can get final consensus result π^*.

4 Experimental Results

We present experiment on UCI dataset iris. The normalized Rand index (R_n) [6] is adopted. Its value usually range between [0,1]. The higher value, indicate that the higher quality of clustering. We demonstrate the cluster validity of WIKCC by comparing it with two well-known consensus clustering algorithms the K-means-based algorithm (KCC) [2], the hierarchical algorithm (HCC) [5].

4.1 Quality of BPs

We run K-means algorithm 100 times with the initialized number of clusters randomized within [K, $\sqrt{n}$] to generate 100 basic partitionings (BPs) for consensus clustering; K is the true class of data set, n is the number of the instances, the squared Euclidean distance is used for the distance function, the quality of each BPs is measured by R_n, the distribution of quality of BPs is shown as Fig. 2.

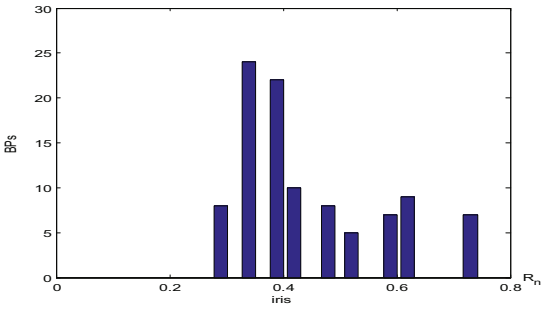

Fig. 2. Clustering quality distribution of BPs

As can be seen in Fig. 2, the distribution of the clustering quality of the BPs show that there is a large proportion of BPs with poor quality, but only quite a small proportion of BPs with relatively high quality. This shows that the incorrect pre-specified number of classes will lead to weak clustering result.

4.2 Exploration of Impact Factors

In order to In order to determine a suitable number of BPs for WIKCC, we explore the influence of the number of BPs on consensus clustering. In the above experiment, r BPs have generated, and r = 100. We randomly select a part of BPs to obtain the subset $\prod^r$, with r = 10, 20, …, 90. For each r we do KCC [2] algorithm 100 times to get 100

consensus clustering result. The distribution of the quality of consensus clustering result for different subset is shown as Fig. 3.

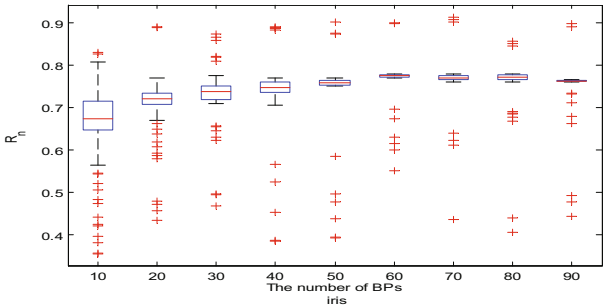

Fig. 3. Impact of the number of BPs to the consensus clustering

As shown in Fig. 3, when $r < 50$, the quality of the consensus result present increasing trend with the increase of r, but when $r > 50$ the result fluctuate in a mall range and nearly tend to be stable, it imply that 50 may be the appropriate number of BPs for WIKCC. Based on above exploration we chose the BPs with the quality of the top 50 BPs for WIKCC.

4.3 WIKCC versus Other Clustering Methods

We compare the WIKCC with KCC and HCC, we choose top 50 better BPs for each method and run on the iris dataset for 10 times.

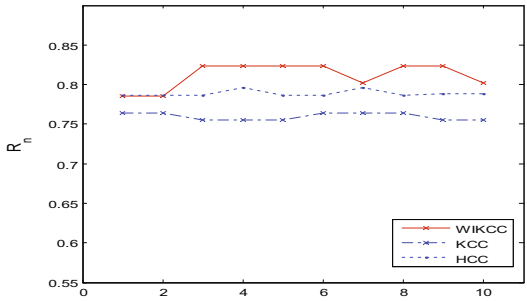

Fig. 4. WIKCC versus KCC and HCC

We can see in Fig. 4. The WIKCC shows significantly higher than KCC, and outperforms better than the HCC in term of the quality of consensus clustering. In addition, comprising the Figs. 2 and 4, we can see that consensus clustering is much better than almost all the basic clustering result obtained by K-means, this indicates that, the consensus clustering method can find the real cluster structure more accurately than a single traditional clustering algorithm by integrating the commonality of many basic

clustering results, so it can obtain a more stable and accurate clustering result by ensemble multiple weak BPS.

5 Concluding Remarks

We explore the influence of the number of BPs on the consensus clustering and chose appropriate number better BPs for WIKCC. The weight is designed by the NMI method between two BPs based on co-occurrence matrix. Finally, the experiment on iris demonstrates that WIKCC outperforms the state-of-the-art well-known KCC and HCC algorithms in terms of clustering quality. In the future, we will explore the more other factors that have influence on the performance of KCC, and we will consider more other factors when designing the weights.

Acknowledgment. Projected supported by National Natural Science Foundation of China (61472231, 61170038, 61502283, 61640201), Jinan City independent innovation plan project in College and Universities, China (201401202), Ministry of education of Humanities and social science research project, China (12YJA630152), Social Science Fund Project of Shandong Province, China (11CGLJ22, 16BGLJ06).

References

1. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**(12), 583–617 (2002)
2. Wu, J., Liu, H., Xiong, H., et al.: K-means-based consensus clustering: a unified view. IEEE Trans. Knowl. Data Eng. **27**(1), 155–169 (2015)
3. Yu, Z., Luo, P., You, J., et al.: Incremental semi-supervised clustering ensemble for high dimensional data clustering. IEEE Trans. Knowl. Data Eng. **28**(3), 701–714 (2016)
4. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080. ACM (2009)
5. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)
6. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)