Ensemble Clustering via Random Walker Consensus Strategy

D. D. Abdala, P. Wattuya, and X. Jiang

Department of Computer Science, University of Münster, Germany
{abdala, wattuya, xjiang}@math.uni-muenster.de

Abstract—In this paper we present the adaptation of a random walker algorithm for combination of image segmentations to work with clustering problems. In order to achieve it, we pre-process the ensemble of clusterings to generate its graph representation. We show experimentally that a very small neighborhood will produce similar results if compared with larger choices. This fact alone improves the computational time needed to produce the final consensual clustering. We also present an experimental comparison between our results against other graph based and well known combination clustering methods in order to assess the quality of this approach.

Keywords—ensemble clustering; random walker.

I. INTRODUCTION

Clustering combination has emerged as a valid option in data clustering. It is an elegant way to deal with the problem of choosing the fittest clustering result in cases where little or nothing is known about the data set. It also works as a way to smooth the final result when different clusterings can potentially present dissimilar partitionings. Finally, it is also a valid way to improve the final result by gathering correct evidence among all the clusterings and merging it in a final consensual result. There is a number of methods already published addressing this topic.

One of the most popular is the median partition (MP) formulation [1]. It can be formally stated as follows: Given $M$ clusterings $C_1, \cdots, C_M$ over a set $P$ of $N$ input patterns and $d(\cdot, \cdot)$, which is a symmetric distance measure between clusterings, find $C^*$ such that:

$$C^* = \arg \min_C \sum_{i=1}^{M} d(C_i, C)$$  \hspace{1cm} (1)

This problem is known to be NP-complete [1], directing the research to the development of heuristics to approximate it. Among the relevant works, Golder and Filkov [2] present a collection of six heuristics. Strehl and Gohsh [3] proposed three graph based heuristics to address the combination problem. In [4] the authors explore the idea of evidence accumulation by combining the clustering results of $M$ clustering results into a co-association matrix. This matrix is later used as a new similarity measure for a standard agglomerative hierarchical clustering algorithm. Finally, Ayad and Kamel [5] proposed three new cumulative voting methods. The problem is formulated as finding a compressed summary of the estimated distribution that preserves the maximum relevance.

In [6] we have presented a combination approach based on a random walker algorithm to fuse multiple image segmentations. In this work we adapt this strategy to address the ensemble clustering problem. The remainder of the paper is organized as follows: Section 2 gives some details of the random walker based image fusion strategy that is needed to understand the remainder of this paper. In section 3 a description of the adaptation to ensemble clustering is presented. Section 4 describes the experiments performed in order to evaluate the validity of the method. Finally, some remarks in Section 5 conclude this paper.

II. RANDOM WALKER ALGORITHM FOR IMAGE SEGMENTATION FUSION

To better explain the changes needed to adapt our image segmentation combination method to deal with general clustering, let’s revisit the original work. Firstly, an ensemble is created during the generation step, followed by the consensus step producing the final consensual result. During the generation step, different clustering algorithms, different initialization parameters, or different views of the data are used in order to create an ensemble of clusterings with sufficient variability. The combination process developed in [6] follows the general consensus clustering model as presented in Figure 1.

![Figure 1. General Combination Clustering Process.](image)

Once the ensemble of results is gathered, a consensus step combines them all into a final consensual result. The consensus step can be divided into 3 parts: a) graph generation; b) seed region generation; and c) ensemble combination.

A. Graph Generation

For our random walker algorithm we need to pre-process the data to generate a graph representation $G(V, E, W)$. 

The generation of the vertices set is straightforward: Simply define a vertex corresponding to each pixel. To generate \( E \), first we need to decide the neighborhood size to be used. For 2D images, two simple choices are available: 4 and 8-neighborhoods. Our algorithm iterates over all vertices computing the edges weights. A weight \( w_{ij} \) indicates how probably the two pixels \( p_i \) and \( p_j \) belong to the same image region. Clearly, this can be guided by counting the number \( m_{i,j} \) of initial segmentations, in which \( p_i \) and \( p_j \) share the same label) and removing all other as a normalization \( p \) indicates how \( p \times m \) constructed from the initial segmentations in \( G \) and \( ref \) refers to how many times the pair \( C \): We build a new graph \( G = G \) of pixels \( K \) and \( p \) is defined as \( w \): The number of can-

\[
w_{ij} = e^{-\beta(1 - \frac{m_{i,j}}{M})}
\]

where \( m_{i,j} \) refers to how many times the pair \( (i, j) \) of pixels occurs in the same cluster among the \( M \) segmentations in the ensemble. We have used \( \beta = 30 \) as a normalization parameter in our tests.

B. Seed Region Generation

Our method applies a two-step strategy to automatically generate seed points:

**Candidate seed region extraction**: We build a new graph \( G^* \) by preserving those edges with weight \( w_{ij} = 1 \) only \( (p_i \) and \( p_j \) share the same label\) and removing all other edges. This step basically retains those edges between two neighboring nodes which are most likely to belong to the same region. Then, we detect all connected subgraphs in \( G^* \) and regard them as a set of initial seeds which are further reduced in the next step.

**Grouping candidate seed regions**: The number of candidate seed regions from the last step is typically higher than the true number of regions in an input image. Thus, a further reduction is performed by iteratively selecting the two candidate seed regions with the highest similarity value and merging them to build one single (possibly spatially disconnected) candidate seed region. Then, the similarity values between a new merged region and all remaining candidate seed regions are recomputed. The similarity \( s(C_i, C_j) \) between candidate seed regions \( C_i \) and \( C_j \) is defined as following:

\[
s(C_i, C_j) = \frac{\sum_{i,j} w_{ij} |(p_i, p_j) \in C_i \times C_j|}{N}
\]

where \( A \) denotes the average of the set \( A \).

C. Ensemble Combination

Given the graph \( G \) constructed from the initial segmentations and \( K \) seed regions, we apply the random walker algorithm [7] to compute the combination segmentation. The computation of random walker probabilities can be exactly performed without the simulation of random walks, but by solving a sparse, symmetric, positive-definite system of equations. Each unseeded pattern is then assigned a \( K \)-
tuple vector, specifying the probability that a random walker starting from that pattern will first reach each of the \( K \) seed regions. A final segmentation is derived by assigning each pixel the label with largest probability. In [6] there are two ways to deal with the final number of target regions: a) specified by the user; or b) automatically determined by the algorithm.

III. RANDOM WALKER FOR ENSEMBLE CLUSTERING

The major change required to adapt the random walker approach above to ensemble clustering problem refers to the graph generation. Since general data sets usually do not present a well structured lattice, the neighborhood of each pattern needs to be determined. The algorithm given in Table 1 shows this process. We specify the neighborhood size \( \delta \) to be used and start by computing the \( \delta \) nearest neighbors for each pattern. We construct an edge between each pattern and its \( \delta \) nearest neighbors. Sometimes two edges may be generated between two vertices \( p_i \) and \( p_j \), one from considering \( p_i \) and the other one considering \( p_j \). In this case, only one of them is added to \( E \). The next step deals with the weight computation. It computes for each edge a weight according to Equation (2).

However, there is no guarantee that \( G \) will be connected. Since the random walker algorithm requires a connected graph, our approach requires a post-process step to guarantee connectedness. This is done by identifying all the connected subgraphs. Given the set of subgraphs a number of options can be adopted in order to produce a final connected graph. Our choice refers to taking each pair of subgraphs \( G_1 \) and \( G_2 \) and inserting an edge between \( p_i \in G_1 \) and \( p_j \in G_2 \) with the smallest distance among all such edges. The newly inserted edges receive weights in the same way as before. Given the constructed connected graph the algorithm previously described can be applied for the ensemble clustering problems with no modifications.

A naive implementation of this graph construction step would require \( O(N(N + \delta M)) \rightarrow O(N^2) \) computational time since \( M \) and \( \delta \ll N \) (see Section 4.1 for a discussion of the neighborhood size \( \delta \)). However, the overall complexity can be improved by smarter ways to compute the \( \delta \) nearest neighbors, for instance as proposed in [8], leading to an
$O(N \log N)$ computational complexity only.

IV. EXPERIMENTAL RESULTS

For our experiments we used eleven UCI data sets [9] as summarized in Table II. Special remarks need to be made about the Mammographic Mass (Mammo) and the Optical Handwritten Digits (Optic) data sets. For Mammo, all patterns with missing values were removed, reducing the data set from 961 to 830 patterns. For the Optic data set we extracted a subset of the first 100 patterns of each digit producing a subset of 1000 patterns. Two additional artificial data sets from [3] were also included in our experiments. The first data set (2D2K) contains 10002D points from two Gaussian clusters and the second data set (8D5K) 1000 points from five multivariate Gaussian distributions (200 points each) in 8D space. These data sets were selected and pre-processed this way in order to make them as compatible as possible to those used in related papers for comparison purpose.

In order to evaluate the final clustering we computed the error rate by matching the clustering results with ground truth information. It is computed by matching the ground truth information available with the consensus clustering obtained. However, the correspondence between them needs to be established. A naive way to do so is to check all possible mapping permutations and then retain the one showing maximum value. A preferable approach to dealing with the permutation problem is to use the Hungarian Algorithm [10] in order to solve the correspondence problem.

A. Assessing the Neighborhood Size

In the first test we have investigated the relationship between the neighborhood size used to create the graph and the error rate. If a small neighborhood could be chosen, this would have a direct positive impact on the overall algorithm performance.

In fact, Figure 2 shows that for all evaluated data sets, the error rate achieved by the random walker algorithm does not fluctuate much with increasing neighborhood sizes. This fact comes as a surprise, since we were expecting data sets with high dimensionality of attributes would require a larger neighborhood in order for the algorithm to perform well. For this reason we fixed $\delta = 6$ for all the following experiments.

B. Assessing the Quality of Final Clustering

In order to assess the quality of the consensus clustering produced by our method we envisioned a comparison with four well known ensemble clustering methods. In Table III E-SL and AL refer to the evidence accumulation approach using single linkage and average linkage proposed in [4]. HGPA (HyperGraph Partitioning Algorithm) and CSPA (Cluster-based Similarity Partitioning Algorithm) were proposed in [3]. All algorithms were executed over ten ensembles of clusterings each composed by twenty runs of K-means (with random initialization, known $K$ and subsets of attributes in order to improve the variability), kernel K-means, hierarchical clustering (based on single, complete, and average linkage), and spectral clustering algorithms. The final ensemble size was $M = 25$. The average results are reported in Table III.

As one can see in Table III, RW performs better for almost all examined data sets if compared to HGPA and CSPA. The only exception is the Optic data set. This is a fair comparison since both methods are graph based. In relation to E-SL/AL, RW always scores equal or better with the only exception of the 8D5K data set. In some cases, e.g. the breast data
set, only a small improvement was obtained. Our algorithm has two advantages over E-SL/AL: 1) it does not impose an additional need in deciding which algorithm version (SL or AL) to use; 2) more importantly, the computation time is minimized.

For smaller data sets (Iris, Wine, Soy, Glass, and Haberman), our algorithm needs a computational time comparable to E-SL/AL. For instance, Iris (150 patterns) was classified in 54 ms by E-SL and 72 ms by E-AL. Our random walker approach required 62 ms. However, once the size of the data set increases, we observe a considerable improvement. For the optic data set (1000 patterns) E-SL took 581 ms, E-AL 1160 ms and RW just 482 ms. As another example, yeast data set (1484 patterns) required 2014 ms (E-SL), 4309 ms (E-AL), and for the RW algorithm, only 746 ms. Note that in our current implementation, the nearest neighbors are found using the naive approach only and a more sophisticated method will further decrease the computational time. The results above were achieved using a fixed number of clusters. However, as described before, the random walker algorithm from [6] has another option of automatically determining this number. We have also tested this option and it turns out that the error rate does not diverge much. This is a very remarkable fact.

A comparison with the method proposed in [5] was not done since only the Breast data set from the UCI database was used there. It is interesting to note that for this data set, that method shows results similar to E-SL/AL. In this particular case, our random walker approach performs slightly better than E-SL/AL.

V. CONCLUSION

In this paper we have shown that our ensemble method, originally designed to be used for combination of image segmentations, can be adapted to handle ensemble clustering with good performance. Random walker based optimization seems to be a valid approach given the fact that it allows the efficient exploration of result spaces in a heuristic way. We have proposed a simple way to compute the graph needed as input to the RW algorithm. Finally, the implications of the neighborhood size to the final quality of the consensus clustering were evaluated. We conclude that a very small neighborhood suffices to produce a good combination quality. We also have identified that the method’s bottleneck is the pre-processing step, more specifically, the graph generation. Our future efforts will be concentrated in the development of a better and faster graph representation.

ACKNOWLEDGEMENTS

Daniel D. Abdala thanks the CNPq, Brazil-Brasilia for granting him a Ph.D. scholarship under the process number 290101-2006-9.

REFERENCES


