Active Sampling for Constrained Clustering

Masayuki Okabe
Information and Media Center
Toyohashi University of Technology
Email: okabe@imc.tut.ac.jp

Seiji Yamada
National Institute of Informatics
Email: seiji@nii.ac.jp

***Abstract*—Constrained Clustering is a framework of improving clustering performance by using supervised information, which is generally a set of constraints about data pairs. Since performance of constrained clustering depends on a set of constraints to use, we need a method to select good constraints that are expected to promote clustering performance. In this paper, we propose such a method, which actively select data pairs to be constrained by using variance of clustering iteration. This method consists of a bagging based cluster ensemble algorithm that integrates a set of clusters produced by a constrained k-means with random ordered data assignment. Experimental results show that our method outperforms clustering with random sampling method.**

I. Introduction

Constrained clustering[1], [2] is a kind of semi-supervised learning technique that utilizes labeled and unlabeled data to enhance learning performance. Difference from normal clustering is the use of background knowledge, which is given in the form of constraints about data pairs. Such constraints have two kinds, usually called *must-link* and *cannot-link* constraints. The former is constraints about data pairs that must be in a same cluster, while the latter ones is about data pairs that must be in different clusters. The challenge of constrained clustering to develop utilization methods of such constraints. Some researches proposed to use them in the k-means algorithm as knowledge for assigning data to cluster centers, and some others proposed to use them as constraints for an optimization problem [3].

Although the use of constraints is an effective approach, we have some problems in preparing constraints. One problem is the efficiency of the process. Because constraints must be labeled as "must-link" or "cannot-link" manually by human, his/her cognitive cost seems very high. We need support to help users cut down such an operation cost. The other problem is the effectiveness of the prepared constraints. Many experimental results in recent studies have shown clustering performance does not monotonically improve (sometimes deteriorates) as the number of applied constraints increases. The degree of performance improvement relies on the quality of constraints, not the amount. These results imply that constraints are not all useful, some are effective but some are not effective or even harmful to the clustering. We also need support to help users select only effective constraints that improve the clustering performance. These problems can be resolved by the framework of active learning that automatically selects constraint candidates expected to be useful.

We propose an active sampling method to select a data pair as a constraint candidate that is expected to be useful if true constraint label (must/cannot-link) is given for it. Our method is based on a bagging[4] based cluster ensemble technique and a constrained k-means with random data assignment order. This is a realization of cluster ensemble framework that exploits partially coherent data group from clustering iteration and integrates them into a set of final clusters. Cluster variation can be created by changing data assignment order in a constrained k-means algorithm that is a modified version of COP-Kmeans proposed by Wagstaff[5]. Though original COP-Kmeans algorithm tends to create inconsistent clusters because the results heavily depends on its data assignment order that is generally undecidable, we use such behavior to produce diversity for cluster ensemble.

Once we can produce diversity of clusters, we can measure uncertainty of belongingness of each data pair. Here, belongingness means whether a data pair belongs to a same or not and we can measure such uncertainty by using entropy that is calculated from the probability of the belongingness. It is well known that "uncertainty" is one of major criteria for active learning [6] to select candidates of training data. This is generally called uncertainty sampling [7]. In this research, we need uncertain data pairs that are expected to be useful to improve clustering performance if they are used as constraints. We expect that constraints added by such uncertainty sampling are more effective than constraints selected at random.

The rest of this paper is organized as follows. We first introduce a constrained clustering method based on a bagging framework in Section II. Then we propose our method of active constraint sampling based on the clustering algorithm in Section III. Section IV presents the results of the experiments conducted using six datasets. We finally conclude our work in Section V.

II. Clustering Algorithm

We first propose a constrained clustering algorithm that is based on a bagging based cluster ensemble technique and a constrained k-means with random data assignment order.

Bagging is one of the ensemble learning techniques used to produce a classifier by integrating weak hypotheses generated by a weak learner that has outperforms random classifiers to some extent. Algorithm 1 is our clustering algorithm based on bagging. According to the normal description of the bagging, we can correspond each element of our algorithm as a

978-1-4673-2743-5/12/$31.00 ©2012 IEEE

Algorithm 1 Ensemble of Constrained K-means

INPUT: Dataset $X = \{x_1, ..., x_{|X|}\}$,
Constraints S= $\{(i_1, j_1, y_1), ..., (i_{|S|}, j_{|S|}, y_{|S|})\}$,
k: No. of clusters
OUTPUT: Clusters $C = \{C_1, C_2, ..., C_k\}$

for $t = 1$ to T **do**
  Run constrained k-means procedure(**Algorithm2**). Then, create kernel matrix K^t

$$K^t(i,j) = \begin{cases} 1 & (x_i, x_j) \text{ belongs to same cluster} \\ -1 & (x_i, x_j) \text{ belongs to different clusters} \end{cases}$$

end for

Calculate final kernel matrix K

$$K = \sum_{t=1}^{T} K^t \quad (1)$$

Run kernel k-means algorithm with K, and return final set of clusters C

- Weak Learner → Constrained K-means((Argorithm2))
- Training data → Constraints (must/cannot-link).

A constrained cluster produced by the constrained k-means in each bagging step is used as a kernel matrix. Each element of this kernel matrix indicates whether or not the corresponding data pair belongs to the same cluster. Thus, the kernel matrix represents the link connections of the clusters. From the point of a weak learner, the modified COP-Kmeans predicts the existence of the link between any data pair in the clusters using the constraints as a set of training data. The kernel matrix is an aggregation of these predictions. The kernel matrices are summed up as a kernel. The final clustering result is generated using this final kernel matrix.

Algorithm 2 lays out the entire procedure of our constrained k-means. Although it follows the basic procedure of the standard k-means algorithm, which assigns data to its nearest cluster center, its assignment process is rather complicated since we must consider the weight of constraint. There are mainly two parts to the assignment processes, which consists of the procedure for the constrained and unconstrained data, respectively. The latter one (for the unconstrained data) remains the same as that in a normal k-means process. What we need to consider is the process for the previous one (for constrained data). We must take several cases like those listed below into consideration.

- Whether or not one of the data pairs has already been assigned?
  Our algorithm assigns a constrained data pair at the same time. Since some data contain several constraints, one (or both) of the data may have already been assigned in some cases. We must prepare procedures for such situations.
- Which constraint the data pair has? - must-link or cannot-link?

Algorithm 2 Constrained K-means

INPUT: Dataset X, Constraint Set S, No. of clusters k,
  Weights of Constraints $w_n (n = 1 \sim |S|)$
OUTPUT: Clusters C
Select initial cluster centers
for $r = 1$ to r_{max} **do**
  Assign a random value to each w_n
  Sort constraints in descending order according to w_n^t
  Assign constrained data pairs (x_i, x_j) to cluster centers in sorted order following procedure below
  Let (x_i, x_j) be data pair to be assigned, then each data will be assigned to one cluster centers according to following cases.
  if Both of (x_i, x_j) have not been yet assigned **then**
    Let c_i, c_j be nearest cluster centers for x_i and x_j respectively, then let $d(x_i, c_i)$, $d(x_j, c_j)$ be distances between each data and its nearest cluster center.
    if (x_i, x_j) is constrained by must-link **then**
      if $d(x_i, c_i) < d(x_j, c_j)$ **then**
        Assign x_i and x_j to c_i
      else
        Assign them to c_j
      end if
    else if (x_i, x_j) is constrained by cannot-link **then**
      if $c_i \neq c_j$ **then**
        Assign x_i to c_i, x_j to c_j, respectively
      else
        if $d(x_i, c_i) < d(x_j, c_j)$ **then**
          Assign x_i to c_i, x_j to second nearest center
        else
          Assign x_j to c_j, x_i to second nearest center
        end if
      end if
    end if
  else if x_i has been already assigned and x_j has not been yet assigned **then**
    Let c_i be cluster center to which x_i is assigned
    if x_i and x_j are constrained by must-link **then**
      Assign x_j to c_i
    else if x_i and x_j are constrained by cannot-link **then**
      Assign x_j to cluster center that is nearest to data and is different from c_i
    end if
  else if x_i has not been yet assigned and x_j has already been assigned **then**
    Let c_j be cluster center to which x_j is assigned
    if x_i and x_j are constrained by must-link **then**
      Assign x_i to c_j
    else if x_i and x_j are constrained by cannot-link **then**
      Assign x_i to cluster center that is nearest to data and is different from c_j
    end if
  end if
  Assign rest of data that are not constrained to their nearest cluster centers
  if Clustering result does not change from previous one **then**
    Return result and exit
  else
    Update cluster centers and go to next step
  end if
end for

Depending on the situation described above, we must prepare different procedures according to which constraint the data pair has.

We describe the concrete procedure considering the above cases in Algorithm 2 (l.9 $\sim$ 42).

III. Active Constraints Sampling

The constrained clustering algorithm introduced in previous section produces a variety of clusters. It may occur that a data pair belongs to a same cluster in a bagging step, but they do not in another step. In this way, we can calculate probabilities that a data pair belongs to a same cluster or not from the bagging process. We calculate the entropy for each data pair according to the probabilities and use it as a measurement of "uncertainty" to select constraint candidates for future clustering.

Let (x_i, x_j) be a data pair and p_{ij} be a probability that they belongs to a same cluster. We can calculate a entropy E_{ij} for each (x_i, x_j) as follows.

$$E_{ij} = -p_{ij} \log p_{ij} - (1 - p_{ij}) \log(1 - p_{ij}) \qquad (2)$$

Here, p_{ij} can be estimated from the number of occurrence that (x_i, x_j) belongs to a same cluster during bagging steps $1 \sim T$.

Once we can calculate entropies, we select several data pairs that has higher values as constraint candidates and give them constraint labels (must/cannot-link). This sampling process follows "uncertainty sampling" that is one of major and useful active learning framework. We consider a data pair that has high entropy value to be a good candidate since their constraint label is difficult to predict.

IV. Experiments

We evaluated our proposed method on six datasets. The datasets are summarized in Table I. Glass, wdbc and balance are from the UCI repository[1] and tr11, tr12 and tr23 are from the CLUTO datasets [2].

We compared the following methods for constraint sampling.

- **active**: This is our proposed method, which selects data pairs to be labeled based on the entropy calculated by the clustering sequence obtained during bagging process. Our algorithm starts from 10 (randomly selected) source constraints and repeats active sampling until it gets 100 constraints. In an active sampling, it selects and labels 10 constraint candidates. We repeated this sampling process 10 times and show the average score as the final results.
- **random**: This method randomly selects data pairs to be labeled. We repeated 10 sampling process and show the average score in a similar way as our active method.

Both methods are based on a clustering algorithm described in Section II. The bagging step T was set to 100. Distance metric we used in the experiments was the Euclid distance.

We used normalized mutual information (NMI) to measure the clustering accuracy. The NMI was calculated by using the following formula.

$$\text{NMI}(C, T) = \frac{I(C, T)}{\sqrt{H(C)H(T)}}$$

where C is the set of cluster labels returned by algorithms and T is the set of true cluster labels. $I(C, T)$ is the mutual information between C and T, and $H(C)$ and $H(T)$ are the entropies.

TABLE I
Datasets

	No. of Data	No. of Class	No. of Attribute
glass	214	6	10
wdbc	569	2	30
balance	625	3	4
tr11	414	9	6429
tr12	313	8	5804
tr23	204	6	5832

Fig.1 shows the results on the six datasets. In every graph, horizontal axis indicates the number of constraints used for clustering, and vertical axis indicates the NMI value. Our method slightly outperforms random sampling method in all datasets. Although our method showed little improvement in tr11 and tr23, in other datasets, it outperformed random sampling method and extended the gap as constraints increased. In addition, our method achieved performance improvement in glass and wdbc despite random method did not show any improvement at all.

V. Conclusion

In this paper, we proposed a sampling method for constrained clustering, which actively selects data pairs to be constrained by using variance of clustering iteration. This method consists of a bagging based cluster ensemble algorithm that integrates a sequence of clustering results produced by a constrained k-means with random ordered data assignment. Using this base clustering method, our sampling method measure uncertainty of belongingness of each data pair, which is represented by entropy calculated from the probability of the belongingness. Experimental results showed that our method outperforms clustering with random sampling on six datasets. Though the improvement was mostly slight, it should be noted that our method achieved the performance despite relatively a small set of constraints.

Since active sampling for constrained clustering has not been well developed, our method can be an option. We will investigate the behavior of the method and test it on many other datasets.

[1] http://archive.ics.uci.edu/ml/

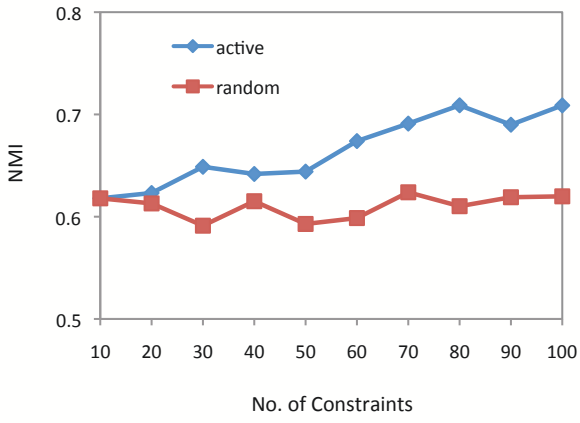
[2] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

References

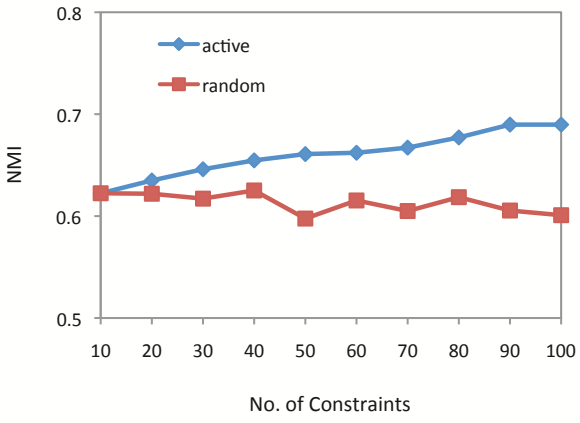
[1] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. The MIT Press, 2006.

[2] S. Basu, I. Davidson, and K. L.Wagstaff, Eds., *Constrained Clustering*. CRC Press, 2008.

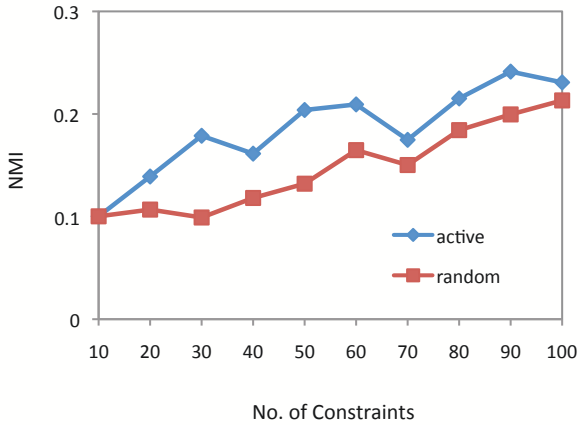
[3] W. Tang, H. Xiong, S. Zhong, and J. Wu, "Enhancing semi-supervised clustering: A feature projection perspective," in *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 707–716.



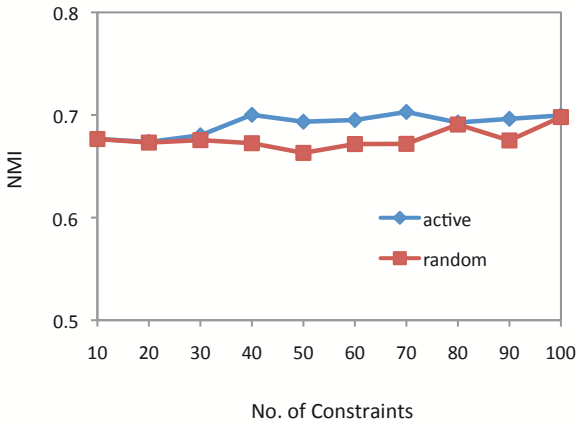
(a) glass



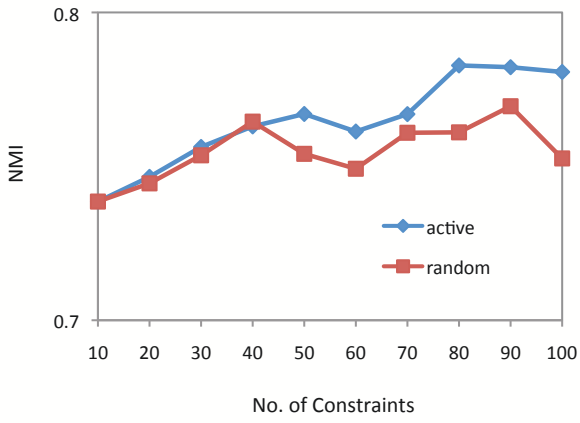
(b) wdbc



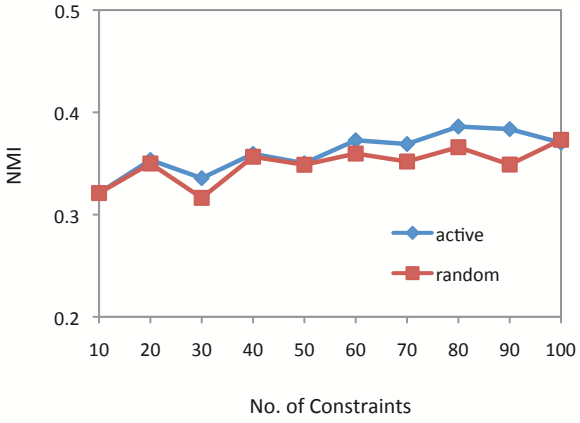
(c) balance



(d) tr11



(e) tr12



(f) tr23

Fig. 1. Results

[4] L. Breiman and L. Breiman, "Bagging predictors," in *Machine Learning*, 1996, pp. 123–140.

[5] K. Wagstaff and S. Roger, "Constrained k-means clustering with background knowledge," in *Proceedings of the 18th International Conference on Machine Learning*, 2001, pp. 577–584.

[6] S. Basu, A. Banjeree, E. Mooney, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04*, 2004, pp. 333–344.

[7] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual ACM SIGIR Conference*, 1994, pp. 3–12.