Paper:

Active Sampling for Constrained Clustering

Masayuki Okabe* **and Seiji Yamada****

*Information and Media Center, Toyohashi University of Technology
1-1 Tempaku, Toyohashi, Aichi 441-8580, Japan
E-mail: okabe@imc.tut.ac.jp
**National Institute of Informatics, SOKENDAI
2-1-2 Chiyoda, Tokyko 101-8430, Japan
[Received October 15, 2013; accepted January 31, 2014]

Constrained clustering is a framework for improving clustering performance by using constraints about data pairs. Since performance of constrained clustering depends on the set of constraints used, a method is needed to select good constraints that promote clustering performance. In this paper, we propose an active sampling method working with a constrained cluster ensemble algorithm that aggregates clustering results that a modified COP-Kmeans iteratively produces by changing the priorities of constraints. Our method follows the approach of uncertainty sampling and measures uncertainty using variations of clustering results where data pairs are clustered together in some results but not in others. It selects the data pair to be labeled that has the most variable result during cluster ensemble process. Experimental results show that our method outperforms random sampling. We further investigate the effect of important parameters.

Keywords: active learning, constrained clustering, cluster ensemble

1. Introduction

Clustering is known to be a basic technique in the field of information retrieval or data mining that is applied to tasks such as image segmentation [1] and topic analysis [2]. There have been vigorous studies to enhance clustering performance by using labeled data pairs in the framework of constrained clustering [3, 4].

Constrained clustering is a semi-supervised learning technique that uses labeled and unlabeled data to enhance learning performance. Constrained clustering is different from normal clustering in the use of constraints about some data pairs where each data item of a pair is constrained about cluster assignment. Such constraints are of two kinds, usually called *must-link* and *cannot-link*. Must-link is a constraint for a data pair that must be in the same cluster, and cannot-link is a constraint for a data pair whose halves must be separated from each other.

Several methods have been proposed to utilize such constraints to improve clustering performance. One method introduced constraint to a *k*-means algorithm, for example, to reduce erroneous data grouping [5], and another method used it to create a better distance measure or kernel matrix [6–10].

Although the use of constraints is an effective approach, we find problems in preparing constraints. One problem is the efficiency of the process. Because constraints must be manually labeled must-link or cannot-link, the user cognitive cost seems very high. We must help users cut down on such labeling cost. Another problem is the effectiveness of prepared constraints. Many experimental results in recent studies have shown that clustering performance does not improve monotonically – and sometimes may even deteriorates – as the number of applied constraints increases [8, 10, 11]. The degree of performance improvement depends on the quality of constraints, not on the quantity. These results imply that constraints are not all useful and that some are effective but others are not effective – or even may be harmful to the clustering. We also must help users to select only effective constraints that improve clustering performance. These problems are solved by the active learning framework [12] that automatically selects constraint candidates expected to be useful.

We propose an active sampling method to select a data pair that is expected to be the most effective if its true constraint label, or must/cannot-link, is given. Our method is based on a bagging-based cluster ensemble technique [13] and constrained *k*-means with a random data assignment order. This realizes a cluster ensemble framework that creates partially coherent data groups from clustering iteration and integrates them into a set of final clusters. Cluster variation is created by changing the data assignment order in a constrained *k*-means algorithm that is a modified version of COP-Kmeans [5]. The original COP-Kmeans algorithm tends to create inconsistent clusters because results depend significantly on its data assignment order, which is generally undecidable, thus we use such behavior to produce diversity for the cluster ensemble.

Once we produce a diversity of clustering results, we observe the diversity of a data pair to be clustered together (or not) during the cluster ensemble process. We regard diversity as an uncertainty sampling measurement [14] that is one of major criteria for active learning [15] to se-

Algorithm 1 Cluster Ensemble

1: INPUT: Data $X = \{x_1, \ldots, x_n\}$,
2: Constraints $S = \{(i_1, j_1, y_1), \ldots, (i_m, j_m, y_m)\}$,
3: Priorities $P = \{p_{(i_1,j_1)}, \ldots, p_{(i_m,j_m)}\}$,
4: Bagging steps T
5: OUTPUT: Clusters $C = \{C_1, C_2, \ldots, C_k\}$
6: **for** $t = 1$ to T **do**
7: Randomize priorities P for constraints S.
8: Run modified COP-Kmeans procedure (**Algorithm 2**).
9: Create kernel matrix K^t from the above clustering result.

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

10: Calculate final kernel matrix K.

$$K = \sum_{t=1}^{T} K^t$$

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

lect data to be labeled. In this research, we propose an active sampling method that follows the approach of uncertainty sampling and measures uncertainty using variations in clustering results where a data pair is clustered together in some results but not in others. It selects the data pair to be labeled that has the most variable results during the cluster ensemble process.

This paper is organized as follows. We first introduce a constrained clustering algorithm that we consider an active sampling method in Section 2. We then propose an active sampling method based on the constrained clustering algorithm in Section 3. Section 4 presents experimental results for six benchmark datasets. We further discuss the effect of important parameters of our proposed method in Section 5. Section 6 presents conclusions.

2. Constrained Clustering Algorithm

In this section, we introduce a constrained clustering algorithm that we consider an active sampling method. This algorithm consists of a bagging-based cluster ensemble technique and modified COP-Kmeans with a random data assignment order.

Algorithm 1 shows the cluster ensemble procedure based on a bagging process that is a well-known ensemble learning technique that aggregates different patterns of predictions produced by a weak learner. In this case, the modified COP-Kmeans works as a weak learner that produces diverse clustering results with some moderate quality. These results are then aggregated in the form of a kernel matrix. Since each K^t is positive semidefinite (Appendix A), linear combination K is also positive semidefinite. We run a kernel k-means using the aggregated kernel matrix to get the final clustering result.

A clustering result produced by the modified COP-Kmeans in each bagging step t is transformed as a kernel matrix K^t in which each element corresponds to a data pair. They are labeled with $+1$ or -1 to indicate whether the corresponding data pair belongs to the same cluster or not. From the point of the bagging framework, the modified COP-Kmeans works as a weak learner to predict a label, or must/cannot-link, of each data pair using constraints as training examples. The quality of a clustering result depends on label prediction accuracy, thus we intend to increase accuracy using bagging that expands the performance of a weak learner.

Algorithm 2 lays out the entire modified COP-Kmeans procedure. Our modified COP-Kmeans algorithm has the following refinements compared to the original one:

- It produces a clustering result regardless of whether it fails to satisfy some constraints.
- It utilizes constraint priorities that are continuously controlled by the bagging process in order to decide the data assignment order in clustering.

The order of data assignment in COP-Kmeans is usually decided at random. Our modified algorithm decides the order by sorting constraint priorities so that a constraint with higher priority is more likely to be satisfied than that with lower priority because the likelihood of constraint satisfaction generally increases if it is considered earlier than other constraints. Our algorithm uses this empirical but practically useful heuristic to decide the order. Although it follows the basic standard k-means algorithm procedure [5], its data assignment process is rather complicated because we introduce the concept of priority for each data pair to produce, in turn, the diversity of clustering results.

There are two main procedures in the modified COP-Kmeans clustering loop. The first is a procedure for assigning unconstrained data to the nearest cluster centers, which follow the standard k-means algorithm. The second is a procedure for assigning constrained data to the nearest cluster centers. This procedure has many conditional branches because we must consider the assignment of two data in a constrained data pair at the same time. To do so, we utilize the priority given for constraints to create clustering diversity. Conditional branches are caused for the following reasons:

- We consider whether each data of a pair has already been assigned. Since some data is restricted by more than one constraint, one or both data may have already been assigned. We must prepare procedures for such conditions.
- We must consider the type of constraint. Depending on the situation above, we prepare different procedures based on the type of constraint the data pair has.

Concrete procedures about such conditions are given in lines 8 to 36 in **Algorithm 2**.

Algorithm 2 modified COP-Kmeans

INPUT: Data X, Constraints S, Priorities P,
    R: No. of maximum repeats in K-means
OUTPUT: Clusters C

Select initial cluster centers
for $r = 1$ to R **do**
  Sort constraints in descending order according to P.
  Assign constrained data pairs (x_i, x_j) to cluster centers in sorted order according to the following procedure.
  if Both of (x_i, x_j) are not assigned yet **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 their nearest cluster centers.
    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.
    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 the second nearest center.
        else
          Assign x_j to c_j, x_i to the second nearest center.
  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.
  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.
  else if Both of (x_i, x_j) are already assigned **then**
    Even if the constraint about (x_i, x_j) is not satisfied, the assignment process will continue.
  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.

3. Active Sampling for Constrained Clustering

We consider an active learning method based on the cluster ensemble introduced in the previous section. The objective of our active learning method is to select a data pair that is expected to promote the most clustering performance if it is labeled. In this research, we propose an active learning method using a concept of uncertainty sampling [14]. Uncertainty sampling is a hypothesis that indicates whether we should label data for which it is the most difficult for a classifier to infer the class. Applying the hypothesis to our case of constrained clustering, we should label the data pair that is the most difficult for our clustering algorithm to infer whether the pair is must-link or cannot-link.

To find such a data pair, we first must decide a criterion to quantify the "uncertainty" produced when our clustering algorithm predicts the label of each data pair. We determine criteria using the cluster ensemble property, namely, the diversity of clustering results. Our clustering algorithm is based on a bagging-based cluster ensemble technique, which utilizes constraints to produce a variety of clustering results. In the cluster ensemble process, a data pair may be clustered together in some results or be assigned to different clusters in other results. We regard such instability as an estimation of uncertainty and quantify behavior using entropy.

Let (x_i, x_j) be a data pair and p_{ij} be the probability that the data pair is grouped in the same cluster in a bagging sequence. We estimate p_{ij} using the number of times that the pair belongs to the same cluster during a sequence of bagging steps.

$$p_{ij} \approx \tilde{p}_{ij} = \frac{\sum_{t=1}^{T} l_{ij}^t}{T} \quad \ldots\ldots\ldots\ldots (1)$$

$$l_{ij}^t = \begin{cases} 1 & \text{if } (x_i, x_j) \text{ is grouped in the same cluster} \\ 0 & \text{if } (x_i, x_j) \text{ is grouped in different clusters} \end{cases}$$

where T is the number of bagging steps.

Using estimate $\tilde{p}_{ij}$, we calculate entropy E_{ij} for each (x_i, x_j) as follows:

$$E_{ij} = -\tilde{p}_{ij} \log \tilde{p}_{ij} - (1 - \tilde{p}_{ij}) \log(1 - \tilde{p}_{ij}) \quad \ldots (2)$$

$(1 - \tilde{p}_{ij})$ is the estimation of the probability that the data pair is grouped in different clusters in a bagging sequence, thus E_{ij} expresses the uncertainty of the data pair to be labeled as must-link or cannot-link.

Once we calculate entropies for all candidate data pairs to be labeled, we sort them in descending order and select several top-ranked pairs from the list, then label them as must-link or cannot-link using true labels, as summarized in **Algorithm 3**.

4. Experiments

4.1. Datasets and Settings

We evaluated our proposed method using the six datasets summarized in **Table 1**. Glass, wdbc and balance are from the UCI repository[1] and tr11, tr12 and tr23

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

Algorithm 3 Active Sampling

INPUT:
  Initial constraints $S = \{(i_1, j_1, y_1), \ldots, (i_m, j_m, y_m)\}$,
  U: No. of maximum repeats in Active sampling,
  N: No. of data pairs to select
for $u = 1$ to U **do**
  Run cluster ensemble procedure (**Algorithm1**). For each data pair (x_i, x_j), calculate entropy E_{ij}.
  Sort data pairs in descending order according to E_{ij}.
  Select N data pairs from the top of the sorted list.
  Label those selected data pairs.
  Add them to S.

Table 1. 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

are from the CLUTO datasets.[2] For CLUTO datasets, we made a feature vector by calculating and normalizing tf·idf for each attribute.

We compared the following methods for constraint sampling:

- *active*: This is our proposed method. Our algorithm starts from 10 randomly selected source constraints and repeats active sampling until it gets 100 constraints. The number of constraints to add at active sampling N is set to 10 in experiments.
- *random*: This method selects data pairs to be labeled at random. Similar to active sampling, it starts from 10 source constraints and repeats random sampling until it gets 100 constraints. Parameter N is also set to 10.

Both methods are based on the constrained clustering algorithm described in Section 2. Bagging step T was set to 100 in both cases. Distance metric we used in experiments was Euclidean distance.

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

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

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

We prepared 10 sets of source constraints. In each dataset, we run 10 trials for both methods by changing source constraints. The final result is the average NMI of 10 trials.

4.2. Results

Figure 1 shows results. The horizontal axis indicates the number of constraints and the vertical axis indicates NMI. Number of bagging steps T was set to 100 in both cases. The number of constraints to be added in an active learning step was set to 10.

Our *active* method showed better or comparable performance compared to random sampling method *random* in all datasets. For the three UCI datasets, *active* clearly outperformed *random*, especially for glass and wdbc, even though *random* showed little performance improvement when increasing constraints. For three CLUTO datasets, although we could not see any significant effect of active sampling, *active* did not show lower performance in any active sampling step.

5. Discussion

Our method has two parameters, N and T, that influence active sampling performance. N is the number of constraints to add in an active sampling step. T is the number of bagging steps. In this section, we investigate the effect of these parameters in more detail.

5.1. Effect of *N*

Figure 2 shows the effect of parameter N, which is the number of constraints to add in an active sampling step. We test the values of parameters 1, 5, 10. A small number of N needs more calculation time, however we expect a better performance improvement because a smaller number of N has a great possibility of avoiding data pairs that are not effective in improving clustering performance under the assumption of uncertainty sampling.

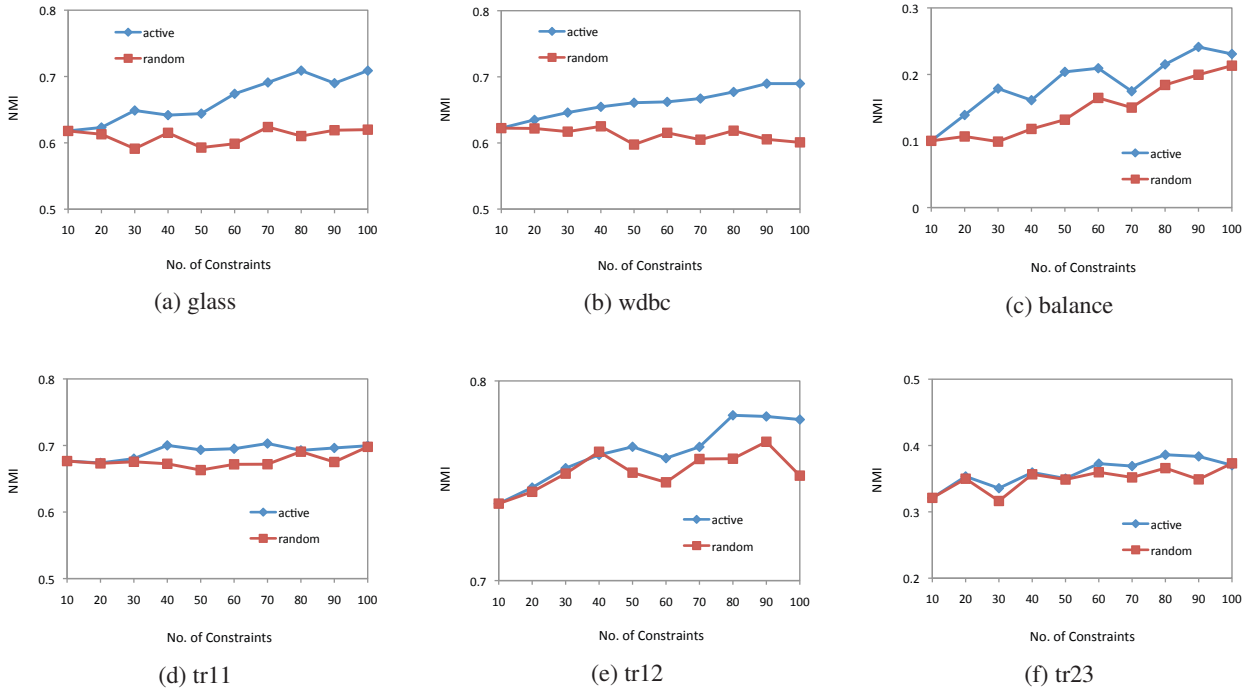
For three UCI datasets, results of $N = 1$ showed the best performance. $N = 5$ results showed no significant improvement, however the number of promising data pairs is very limited in experiments. We did not see any marked change in CLUTO datasets.

5.2. Effect of *T*

Figure 3 shows the effect of parameter T, which is the number of bagging steps. By increasing T, we expect a better estimation of $\tilde{p}_{ij}$ that directly influences the choice of data pairs to be sampled.

Contrary to our expectations, parameter T did not have any significant effect except for the UCI balance dataset. For the balance dataset, the result of $T = 300$ showed the best performance. Since the result of $T = 500$ showed lower performance than $T = 300$, increasing T would not necessarily produce a positive effect.

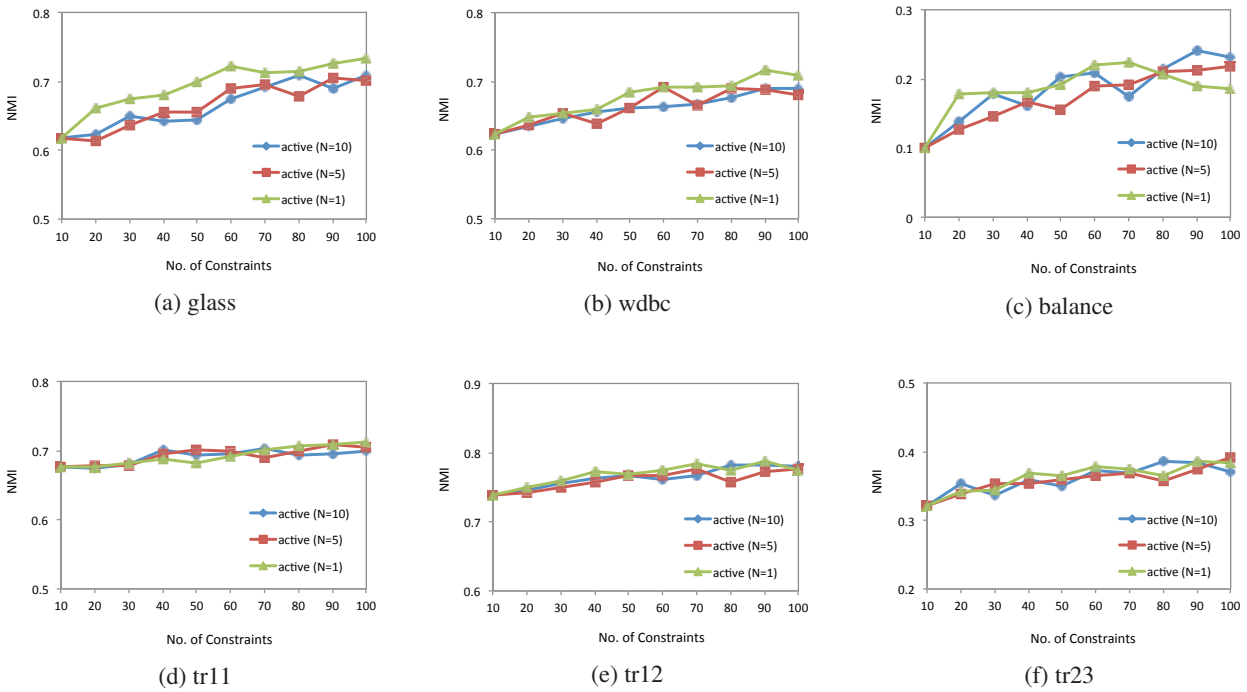
2. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download



(a) glass (b) wdbc (c) balance

(d) tr11 (e) tr12 (f) tr23

Fig. 1. Results ($T = 100$, $N = 10$).



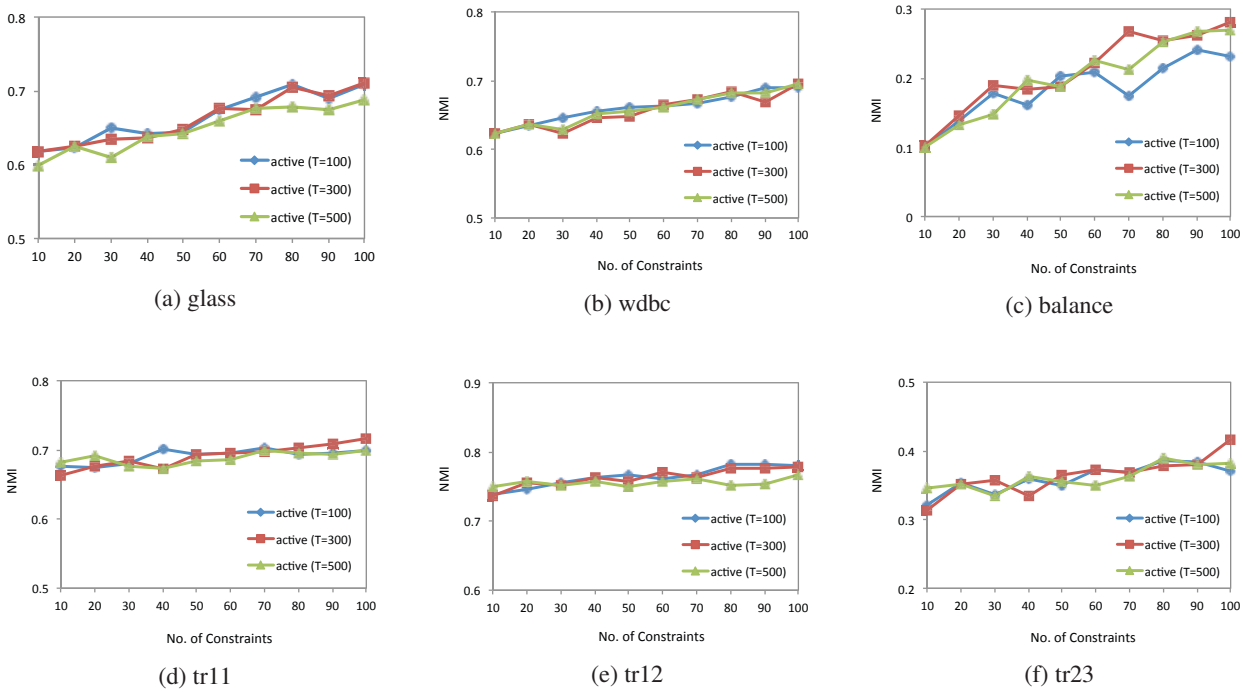
(a) glass (b) wdbc (c) balance

(d) tr11 (e) tr12 (f) tr23

Fig. 2. Effect of number of constraints to add N.

5.3. Entropy of Selected Data Pairs

We used entropy as a measurement for the uncertainty of active sampling and investigated the average entropy for selected data pairs in both *active* and *random*. **Table 2** shows average and standard deviation of entropy for 100 selected data pairs during active sampling using both methods. The average value of *active* is clearly higher than that of *random* in all datasets. In *active*, the value becomes slightly higher as N increases in UCI datasets, however there was no clear difference in CLUTO datasets. This indicates that our proposed method is promising as active sampling technique for constrained clustering even though it is not enough to narrow the numbers of data pair candidates to be selected.



(a) glass (b) wdbc (c) balance

(d) tr11 (e) tr12 (f) tr23

Fig. 3. Effect of bagging steps number T.

Table 2. Selected constraint entropy.

	glass	wdbc	balance	tr11	tr12	tr23
ACTIVE ($N = 10$)	0.99±5.34e-06	0.77±0.09	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00
ACTIVE ($N = 5$)	0.99±8.37e-09	0.80±0.09	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00
ACTIVE ($N = 1$)	0.99±4.54e-10	0.93±0.02	0.99±0.00	0.99±0.00	0.99±0.00	0.99±0.00
RANDOM	0.10±0.06	0.01±0.00	0.04±0.02	0.03±0.01	0.05±0.02	0.14±0.08

6. Conclusions

In this paper, we have proposed an active sampling method for constrained clustering that selects data pairs to be constrained by using variations in clustering iteration. This consists of a bagging-based cluster ensemble algorithm that integrates a sequence of clustering results produced by a modified COP-Kmeans with random ordered data assignment. Using this base clustering, our sampling method measures the uncertainty of the belongingness of each data pair, which is represented by entropy calculated from the probability of belongingness. Experimental results showed that our method outperforms clustering with random sampling for six datasets. Even though improvement was not so significant, it should be noted that our method achieved the required performance despite a relatively small set of constraints.

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

References:

[1] Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.15, pp. 1101-1113, 1993.

[2] L. Dey, A. Mahajan, and S. M. Haque, "Document Clustering for Event Identification and Trend Analysis in Market News," In Proc. of the 2009 Seventh Int. Conf. on Advances in Pattern Recognition, ICAPR'09, pp. 103-106, IEEE Computer Society, 2009.

[3] S. Basu, I. Davidson, and K. L.Wagstaff (Eds.), "Constrained Clustering," CRC Press, 2008.

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

[5] K. Wagstaff and S. Roger, "Constrained K-means Clustering with Background Knowledge," In Proc. of the 18th Int. Conf. on Machine Learning, pp. 577-584, 2001.

[6] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," In Proc. of the 24th Int. Conf. on Machine learning, pp. 361-368, 2007.

[7] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman, "Online Metric Learning and Fast Similarity Search," In Proc. of Neural Information Processing Systems, pp. 761-768, 2008.

[8] D. Klein, S. D. Kamvar, and C. D. Manning, "From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering," In Proc. of the 19th Int. Conf. on Machine Learnning, pp. 307-314, 2002.

[9] Z. Li, J. Liu, and X. Tang, "Pairwise constraint propagation by semidefinite programming for semi-supervised classification," In Proc. of the 25th Int. Conf. on Machine learning, pp. 576-583, 2008.

[10] W. Tang, H. Xiong, S. Zhong, and J. Wu, "Enhancing Semi-Supervised Clustering: A Feature Projection Perspective," In Proc. of the 13th Int. Conf. on Knowledge Discovery and Data Mining, pp. 707-716, 2007.

[11] Y. Liu, R. Jin, and A. K. Jain, "BoostCluster: boosting clustering by pairwise constraints," In Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, KDD'07, pp. 450-459, New York, NY, USA, 2007.
[12] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," J. of Machine Learning Research, Vol.2, pp. 45-66, March 2002.
[13] L. Breiman and L. Breiman, "Bagging Predictors," In Machine Learning, pp. 123-140, 1996.
[14] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," In Proc. of the 17th Annual ACM SIGIR Conf., pp. 3-12, 1994.
[15] S. Basu, A. Banjeree, E. Mooney, A. Banerjee, and R. J. Mooney, "Active Semi-Supervision for Pairwise Constrained Clustering," In In Proc. of the 2004 SIAM Int. Conf. on Data Mining (SDM-04), pp. 333-344, 2004.

Appendix A. Proof of K^t as Positive Semidefinite

K^t is a n-dimensional square matrix. Suppose indices of the row (or column) of matrix K^t are aligned for members of clusters. That is, members of cluster 1 are aligned with indices from 1 to l_1 in row K^t and members of cluster 2 are aligned with indices from l_1+1 to l_1+l_2, etc. Here, l_i is the number of members in the i-th cluster.

K^t is represented as follows:

$$K^t = 2\begin{pmatrix} C_1 & O & O & O \\ O & C_2 & O & O \\ & & \ddots & \\ O & O & & O \\ O & O & O & C_k \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & \cdots & 1 \\ & & & \\ 1 & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}.$$

where C_i is a square matrix whose dimension is equal to the number of members in the i-th cluster and elements are all 1. O is a matrix whose elements are all 0, thus elements except for C_i in K^t are all zero. Elements of the second matrix in the above formula are all 1.

Letting A and B be the first and the second matrix in the above formula, that is, $K^t = 2A - B$, K^t is positive semidefinite if A and B are positive semidefinite because K^t is a linear combination of these matrices.

For any vector $\mathbf{z} = \{z_1, z_2, \ldots, z_n\}$, we calculate $\mathbf{z}A\mathbf{z}$ as follows:

$$\begin{aligned} \mathbf{z}A\mathbf{z} &= \mathbf{z}C_1\mathbf{z} + \mathbf{z}C_2\mathbf{z} + \cdots + \mathbf{z}C_k\mathbf{z} \\ &= \left\{\sum_{i=1}^{l_1} z_i\right\}^2 + \left\{\sum_{i=l_1+1}^{l_1+l_2} z_i\right\}^2 + \cdots + \left\{\sum_{i=n-l_k+1}^{n} z_i\right\}^2 \\ &\geq 0. \end{aligned}$$

A is positive semidefinite. We prove that B is positive semidefinite in a similar way.

$$\begin{aligned} \mathbf{z}B\mathbf{z} &= \left\{\sum_{i=1}^{n} z_i\right\}^2 \\ &\geq 0 \end{aligned}$$

Since both A and B are positive semidefinite, K^t is also positive semidefinite.

Name:
Masayuki Okabe

Affiliation:
Toyohashi University of Technology

Address:
1-1 Tenpaku, Toyohashi, Aichi 441-8580, Japan
Brief Biographical History:
2001- Researcher of JST, Kyoto University
2003- Research Associate, Toyohashi University of Technology
Main Works:
• machine learning
• information retrieval
• data mining
Membership in Academic Societies:
• The Japanese Society for Artificial Intelligence (JSAI)

Name:
Seiji Yamada

Affiliation:
National Institute of Informatics, Graduate University for Advanced Studies (SOKENDAI)

Address:
2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430, Japan
Brief Biographical History:
1989- Research Associate, Osaka University
1991- Lecturer, Osaka University
1996- Associate Professor, Tokyo Institute of Technology
2002- Professor, National Institute of Informatics / SOKENDAI
Main Works:
• human-agent interaction
• intelligent interactive systems
• interactive machine learning
Membership in Academic Societies:
• Association for Computing Machinery (ACM)
• The Institute of Electrical and Electronics Engineers (IEEE)
• Association for the Advancement of Artificial Intelligence (AAAI)
• The Japanese Society for Artificial Intelligence (JSAI)