2013 Conference on Technologies and Applications of Artificial Intelligence

Uncertainty Sampling for Constrained Cluster Ensemble

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

Seiji Yamada
Digital Content and Media Sciences Research Division
National Institute of Informatics
Tokyo, Japan
Email: seiji@nii.ac.jp

***Abstract*—Constrained Clustering is a framework of improving clustering performance by using 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 selects data pairs to be constrained by using variance of clustering iteration. This method consists of a boosting based cluster ensemble algorithm that integrates a set of clusters produced by a constrained k-means with controlled data assignment order. Experimental results show that our method outperforms clustering with random sampling method.**

***Keywords*-active learning; constrained clustering; uncertainty sampling; cluster ensemble;**

I. Introduction

Clustering is a basic data mining technique to find similar data groups in a dataset. We can execute clustering algorithms by giving feature vectors and a similarity measure. However, it sometimes happens that similarity measure does not fit the target dataset and get unsatisfied results.

Constrained clustering [1], [2] can be applied to such a situation. It is a kind of semi-supervised learning technique that utilizes labeled and unlabeled data to enhance learning performance. Constrained clustering is different from normal clustering in the use of background knowledge that 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. According to this framework, users can modify or fix the problem of pre-given similarity measure by giving such constraints. For example, interactive image segmentation [3], [4] is a typical task for constrained clustering. Depending on the segmentation performance, users can give appropriate constraints to get better results.

Although there have been proposed several constrained clustering methods [5], [6], [7], [8], 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 reduce such 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 rather than the amount. These results imply that constraints are not all useful, some are effective but some are not effective or even harmful to clustering. We also need support to help users select only effective constraints that improve clustering performance. These problems can be resolved by the framework of active learning that automatically selects constraint candidates expected to be useful.

In this paper, we propose an active learning 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. We realize an uncertainty sampling based active learning method on a constrained cluster ensemble algorithm that integrates a sequence of clustering results produced by constrained k-means using boosting framework. The cluster ensemble is based on a boosting [9] framework with a constrained k-means algorithm. Controlling the priorities of constraints by boosting, the constrained k-means that is a modified version of COP-Kmeans [10] produces the variation of clustering results by changing the data assignment order.

We exploits this variation to measure the uncertainty for a data pair to be must-link or cannot-link. It is well known that “uncertainty” is one of major criteria for active learning [11] to select candidates of training data. This is generally called uncertainty sampling [12]. 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 calculate the uncertainty using two measurements. One is the number that a data pair belongs to the same or different cluster during cluster ensemble process. The other is the kernel value that is produce for a data pair by boosting process. We expect that constraints selected by using uncertainty sampling are more effective than constraints selected at random.

The rest of this paper is organized as follows. We first introduce our basic constrained cluster ensemble algorithm with boosting and constrained k-means in Section II. Then

978-1-4799-2528-5/13 $31.00 © 2013 IEEE
DOI 10.1109/.56

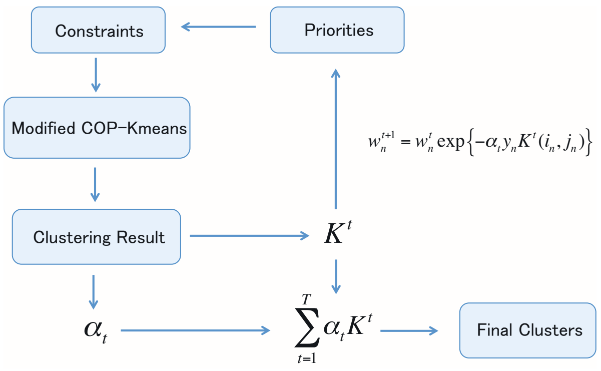


Figure 1. Constrained Cluster Ensemble

we propose our uncertainty sampling based active learning method for constrained clustering in Section III. Section IV presents the results of the experiments on six datasets from UCI repository and public shape datasets. We finally conclude our work in Section V.

II. Constrained Cluster Ensemble

In this section, we briefly explain a constrained clustering algorithm on which we consider active learning.

The constrained clustering algorithm is based on a cluster ensemble approach, where a certain number of slightly or significantly different clustering results are integrated into a set of clusters. One of the important things in cluster ensemble is how to create such difference. While there are so many mechanisms to produce different patterns of clustering, we adopt a boosting based method [13] that is appropriate for our "constrained" cluster ensemble.

This method is based on a boosting technique and a constrained k-means algorithm. Boosting is one of the ensemble learning techniques used to produce a classifier by integrating weak hypotheses generated by a weak learner that outperforms random classifiers to some extent. The boosting process mainly follows the AdaBoost algorithm[9], which is a well-known framework to enhance the classifier ability by flexibly changing the weights of the training data. Figure 1 illustrates how our clustering algorithm works according to the AdaBoost procedure. Although we normally use boosting for classification learning that is different from constrained clustering, we can naturally apply by finding correspondences as follows.

- Weak Learner → Constrained k-means
- Training data → Constraints (must/cannot-link)
- Weights for training data → Weights for constraints.

Once given a set of constraints, the constrained k-means algorithm that works as a weak learner for the boosting runs and produces a clustering result. The clustering result produced in each boosting step is transformed into a kernel matrix K^t. 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 constrained k-means 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. As boosting step goes on, constrained k-means produces different clustering results. They are finally summed up as a kernel matrix with their importance values α_t that are calculated by using the error rate of the constraints satisfaction in each boosting step. The final clustering result is generated using this final kernel matrix.

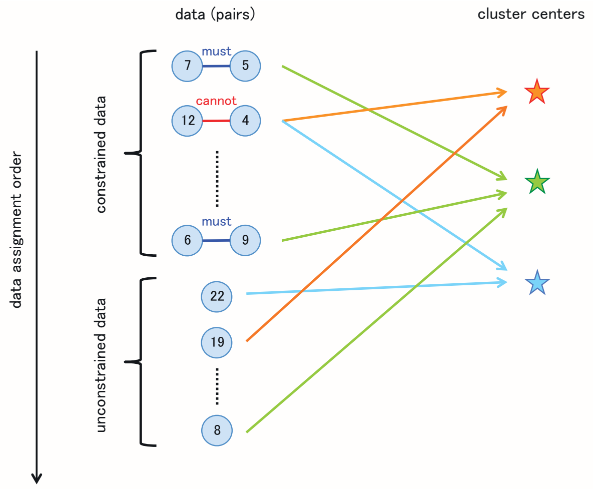


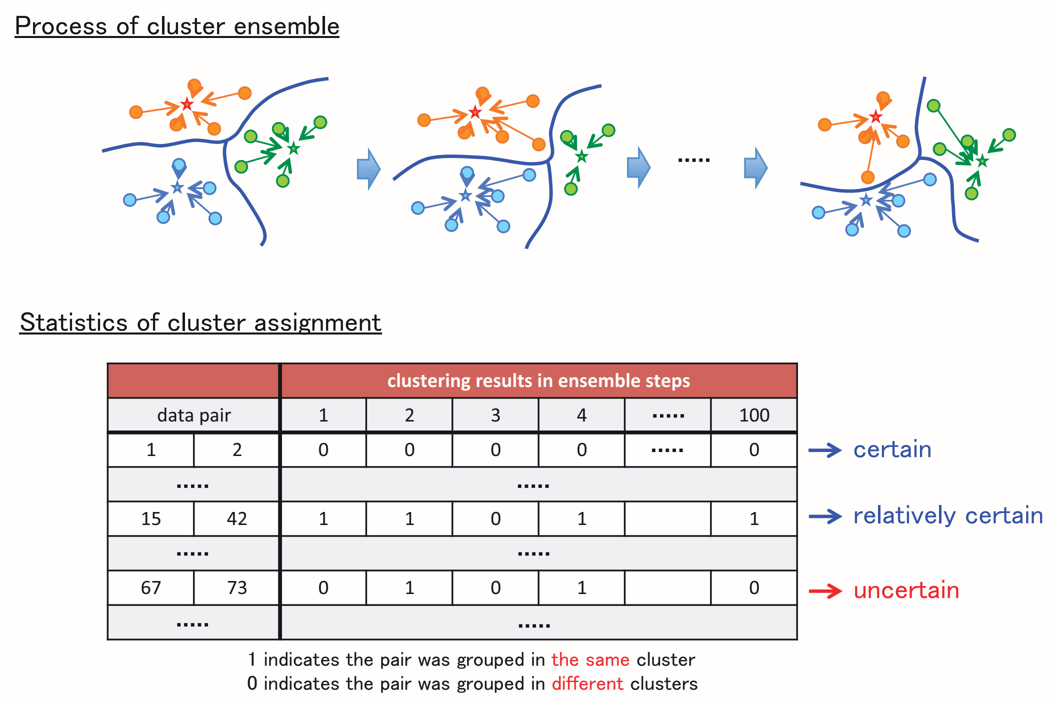
Figure 2. Constrained K-means

Another important role of the boosting framework is to give a priority w_n^t to each constraint. The priority w_n^{t+1} for nth constraint at $t+1$th boosting step is calculated by the following formula.

$$w_n^{t+1} = w_n^t \exp(-\alpha_t y_n K^t(i_n, j_n))$$

where y_n is the label for nth constraint and $K^t(i_n, j_n))$ is the kernel value for nth constraint. The constraints that are not satisfied by the weak learner will be given higher priorities in order to be satisfied in the next round. These priorities are used to control the data assignment order in the constrained k-means. The constraints with higher priorities will be satisfied earlier than those with low priorities. The priority calculation can be dealt with by changing the coefficients of the loss function in the boosting.

Figure 2 illustrates how constrained k-means uses the priorities for constraints. The constrained k-means algorithm is developed from the COP-Kmeans [10] by modifying to make work as a weak learner for constrained cluster ensemble. Although it follows the basic procedure of the standard k-means algorithm that assigns data to its nearest cluster center, its assignment process is rather complicated since we must consider the weight of constraints. There are



		clustering results in ensemble steps						
data pair		1	2	3	4	·····	100	
1	2	0	0	0	0	·····	0	→ certain
·····		·····						
15	42	1	1	0	1		1	→ relatively certain
·····		·····						
67	73	0	1	0	1		0	→ uncertain
·····		·····						

1 indicates the pair was grouped in the same cluster
0 indicates the pair was grouped in different clusters

Figure 3. Variability of the Clustering Results

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

III. Uncertainty Sampling for Constrained Cluster Ensemble

Based on the constrained cluster ensemble algorithm, we consider active learning that tries to infer the most promising constraint in terms of improving clustering performance the best. In this section, we propose an active learning technique that is based on the uncertainty sampling [12]. Uncertainty sampling is based on a hypothesis that we should label the data which is the most difficult to infer. According to the hypothesis, we should label the data pair that is the most difficult to infer that it is must or cannot-link. In order to apply uncertainty sampling, we first have to decide the criteria to quantify "uncertainty". Our criteria is based on two types of variability of the clustering results that are produced during the boosting process.

The first criteria is the variability of the number that a data pair belong to the same or different cluster during the boosting process. As illustrated in Figure 3, the constrained clustering algorithm introduced in previous section produces a variety of clusters. It may occur that a data pair belongs to the same cluster in a boosting step, but they do not in another step. In this way, we can calculate probabilities that a data pair belongs to the same cluster or not from the boosting process. We measure the variability by calculating the entropy for each data pair according to the probabilities and use it as the first criteria 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 an entropy E_{ij} for each (x_i, x_j) as follows.

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

Here, p_{ij} can be estimated from the number of occurrence that (x_i, x_j) belongs to the same cluster during boosting 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). We consider a data pair that has high entropy value to be a good candidate since its constraint label is difficult to predict.

The second criteria is the variability of the final kernel value that is produced for a data pair in each boosting step. As we described before, clustering results are transformed and integrated into a kernel matrix whose element indicates the strength of the corresponding data pair is a must or cannot-link. Thus if the kernel value is positively large, we can infer it is must-link. In the same way, if the kernel value is negatively large, we can infer it is cannot-link. On the other hand, we can also consider whether a data pair is must or cannot-link is uncertain if the absolute kernel value is small. Since the final kernel value is the sum of positive and negative values produced during each boosting step, smallness of the absolute kernel value can be used as a measurement of "uncertainty".

Using above two types of measurement, we calculate the value of uncertainty U as follows.

$$U_{ij} = E_{ij}/K_{ij} \tag{2}$$

where i and j means each index of a data pair, and K is a kernel matrix. We calculate U_{ij} for any unlabeled data pair and select a data pair that has the largest value.

IV. Experiments

A. Datasets

We evaluated our proposed method on six datasets. The datasets are summarized in Table I. Iris, Glass, Ecoli, Wdbc are from the UCI repository[1] and Pathbased and Spiral are from a public shape datasets for clustering [2]. We rescaled the range of attribute values of each data [3] in four datasets from UCI repository to avoid negative effect of attribute scale unbalance. For two shape datasets, we applied RBF kernel with local scaling[14] because they are difficult to be clustered in Euclidean space. We set 8 to the number of nearest neighbors for local scaling.

B. Methods and Evaluation

We compared the following methods for constraint sampling.

- **ACTIVE**: This is our proposed method, which selects data pairs to be labeled using two types of "uncertainty" measurement described in the previous section. Our algorithm starts from 100 (randomly selected) source constraints and repeats uncertainty sampling until it gets 200 constraints. Since constrained clustering algorithms generally need a certain number of constraints to produce their effect, we starts from 100 souce constraints. In each sampling step, it selects one data pair to be labeled. We repeated this sampling process and show the average score as the final results. The number of maximum boosting steps T was set to 100.
- **RANDOM**: This method randomly selects data pairs to be labeled. We repeated the sampling process and show the average score in a similar way as our active method.

Table I
Datasets

	No. of Data	No. of Class	No. of Attribute
Iris	150	3	4
Glass	214	6	10
Ecoli	336	8	7
Wdbc	569	2	30
Pathbased	300	3	2
Spiral	312	3	2

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.

C. Results

Figure 4 shows the results of the datasets. In each graph, horizontal axis indicates the number of constraints used for constrained cluster ensemble, and vertical axis indicates the NMI value.

As for UCI datasets, ACTIVE outperforms RANDOM on Iris and Glass datasets. In Glass, the difference of NMI expands as the number of constraints increases. For other two datasets Ecoli and Wdbc, ACTIVE shows almost comparable results even if it gets more constraints. Shape datasets also showed similar results. ACTIVE is better than RANDOM on Pathbased, but comparable on Spiral.

V. Conclusion

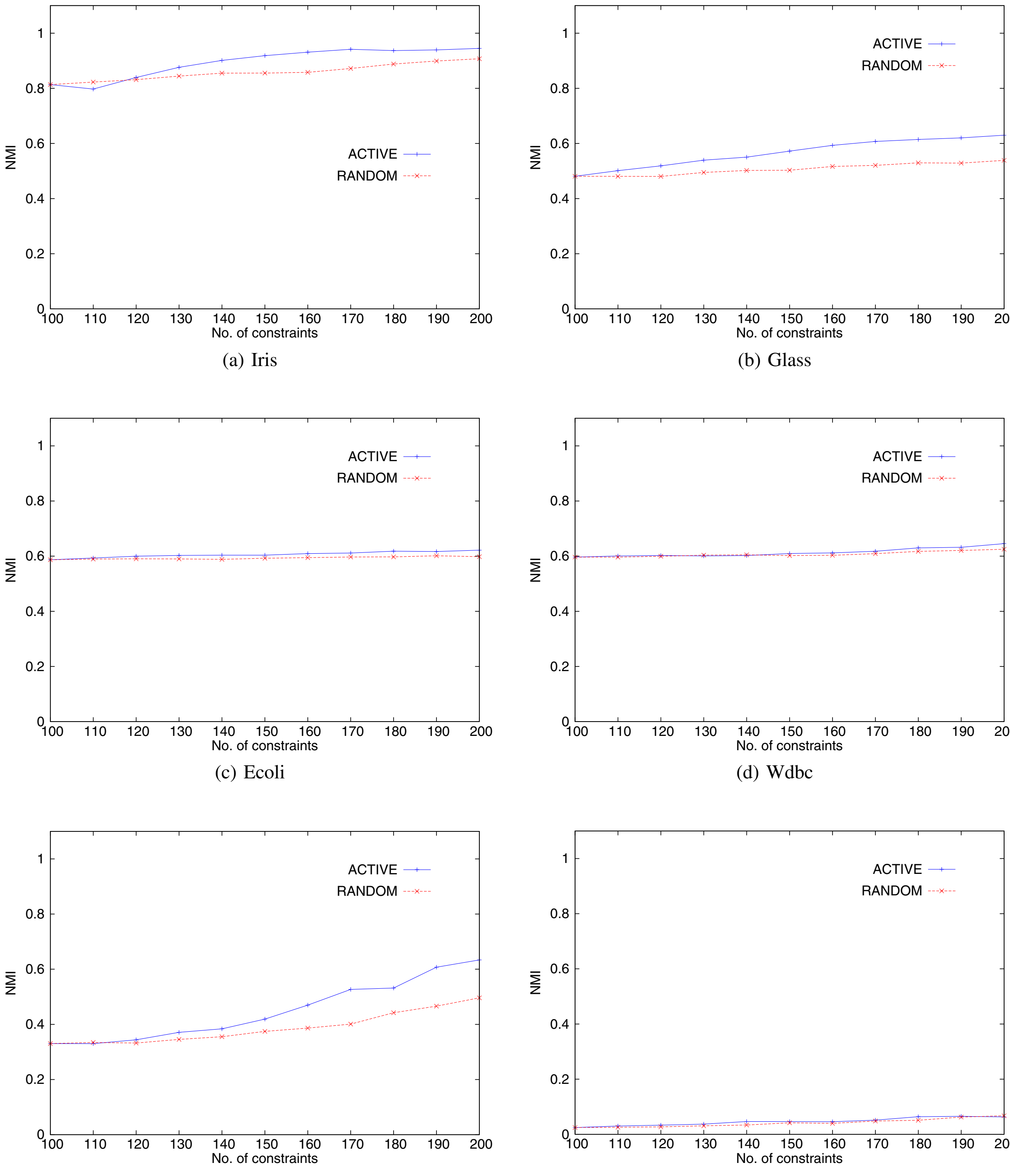
In this paper, we proposed an active learning method for constrained clustering, which actively selects data pairs to be constrained. Our active learning method is a realization of uncertainty sampling on a constrained cluster ensemble algorithm that integrates a sequence of clustering results produced by constrained k-means using boosting framework. We utilize the variability of the clustering results through the cluster ensemble process to measure the uncertainty for any data pair to be must or cannot-link. The uncertainty is calculated by using two measurements. One is the number that a data pair belongs to the same or different cluster during cluster ensemble process. The other is the kernel value that is produce for a data pair by boosting process.

Experimental results showed that our method outperforms or is comparable to a random sampling method on six datasets. Though the effectiveness depends on data and basic performance of our cluster ensemble method, we verified

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

[2] http://cs.joensuu.fi/sipu/datasets/

[3] $v' = \frac{v-\min\{v\}}{\max\{v\}-\min\{v\}}$ (v': rescaled attribute value)



(a) Iris

(b) Glass

(c) Ecoli

(d) Wdbc

(e) Pathbased

(f) Spiral

Figure 4. Results

uncertainty sampling approach has potential to work well on some datasets.

Since active learning 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.

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] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *ACM Transactions on Graphics (TOG)*, vol. 1, no. 212, 2004, pp. 309–314.

[4] Y. Boykov and M. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images,” in *ICCV*, no. July, 2001, pp. 105–112.

[5] W. Tang, H. Xiong, S. Zhong, and J. Wu, “Enhancing semi-supervised clustering: A feature projection perspective,” in *KDD*, 2007, pp. 707–716.

[6] Q. Xu, M. desJardins, and K. L. Wagstaff, “Active constrained clustering by examining spectral eigenvectors,” in *Proceedings of the 8th International Conference on Discovery Science*, 2005, pp. 294–307.

[7] S. C. H. Hoi, R. Jin, and M. R. Lyu, “Learning nonparametric kernel matrices from pairwise constraints,” in *ICML*, 2007, pp. 361–368.

[8] L. Wu, R. Jin, S. C. H. Hoi, J. Zhu, and N. Yu, “Learning bregman distance functions and its application for semi-supervised clustering,” in *NIPS*, 2009, pp. 2089–2097.

[9] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *EuroCOLT*, 1995, pp. 23–37.

[10] K. Wagstaff and S. Roger, “Constrained k-means clustering with background knowledge,” in *ICML*, 2001, pp. 577–584.

[11] S. Basu, A. Banjeree, E. Mooney, A. Banerjee, and R. J. Mooney, “Active semi-supervision for pairwise constrained clustering,” in *SDM*, 2004, pp. 333–344.

[12] D. Lewis and W. Gale, “A sequential algorithm for training text classifiers,” in *SIGIR*, 1994, pp. 3–12.

[13] M. Okabe and S. Yamada, “Clustering by learning constraints priorities,” in *ICDM*, 2012, pp. 1050–1055.

[14] P. Perona and L. Zelnik-Manor, “Self-Tuning Spectral Clustering,” in *Advances in Neural Information Processing Systems 17*, vol. 2, 2005, pp. 1601–1608.