An Interactive Tool for Human Active Learning in Constrained Clustering

Masayuki Okabe
Toyohashi University of Technology
Tempaku 1-1, Toyohashi, Aichi, Japan
okabe@imc.tut.ac.jp

Seiji Yamada
National Institute of Informatics
Chiyoda, Tokyo, Japan
seiji@nii.ac.jp

Abstract— **This paper describes an interactive tool for constrained clustering that helps users to efficiently select effective constraints during the constrained clustering process. Constrained clustering is a promising technique for smart data aggregation or filtering, which is indispensable for the user activity on the Web. Effective bias is necessary for the constraints selection in order to make it a more practical technique, We approach this problem by incorporating human biasing using an easy manipulatable interactive tool. This tool has several functions such as the 2-D visual arrangement of a dataset and constraint assignment by mouse manipulation. Moreover, it can be used to execute distance metric learning and k-means clustering. In this paper, we show an overview of the tool and how it works, especially for the functions for display arrangement by using multi-dimensional scaling and incremental distance metric learning. In the experiments, we investigated the performance of the sampling heuristics found by observing the interaction between the users and our tool. The results show that the heuristic outperforms the random sampling method both in the two benchmark datasets from the UCI repository and a Web page dataset from the Open Directory Project.**

I. Introduction

Constrained clustering is a promising approach for improving the accuracy of clustering by using some prior knowledge about the data. We generally use two types of simple constraints about a pair of data as the prior knowledge. The first constraint is called "must-link", which is a pair of data that must be in the same cluster. The second one is called "cannot-link", which is a pair of data that must be in different clusters. Several approaches have been proposed that use these constraints. For example, a well-known constrained clustering algorithm called COP-Kmeans [1] uses these constraints as the exceptional rules for the data allocation process in a k-means algorithm. The data may not be allocated to the nearest cluster center if the data and a member of the cluster form a cannot-link, or the data and a member of the other cluster form a must-link. Other studies [2], [3], [4] are based on a supervised metric learning that uses the constraints to modify an original distance (or called "similarity", "kernel") matrix to satisfy the target distance or value of each constraint. In addition, a hybrid method [5] has been proposed.

Although the use of constraints is an effective approach, we have some problems in preparing constraints. One problem is the efficiency of the process. Since a human user generally needs to label many constraints with "must-link" or "cannot-link", his/her cognitive cost seems very high. Thus, we need an interactive system that can 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 that the 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 the constraints very much. These results imply that not all constraints are useful, some are effective but some are ineffective or even harmful to the clustering. We also need an interactive system to help users select suche effective constraints that improve the clustering performance. The second problem is much more related to an active learning that has not been thus far researched in the field of constrained clustering.

There have been various studies on Web page clustering and Web usages clustering [6]. We took this work into consideration for interactive clustering as is one of the promising approaches for application to such large-scaled clustering. Thus, we think this work will provide fundamental technologies for Web clustering.

We approach these problems by developing an interactive tool that helps users efficiently select effective constraints during the clustering process. The main objectives to build the interactive tool can be sum up as follows.

1) To provide an interactive environment in which users can visually recognize the proximity of data, and give constraints easily by mouse manipulation.
2) To provide hints for the better selection strategies through the interaction process between the interactive system and users.

In addition to the 2-D visual arrangement of a dataset and the constraint assignment function, our prototype tool has distance metric learning and k-means clustering that can be quickly executed as the background process. Using these functions, the users can compare the results of the clustering before and after the constraints addition easily. We consider such interactions helpful for providing hints for better selection strategies.

Although there are many datamining tools that have clustering function, we have not found any other tool

doi:10.4304/jetwi.3.1.20-27

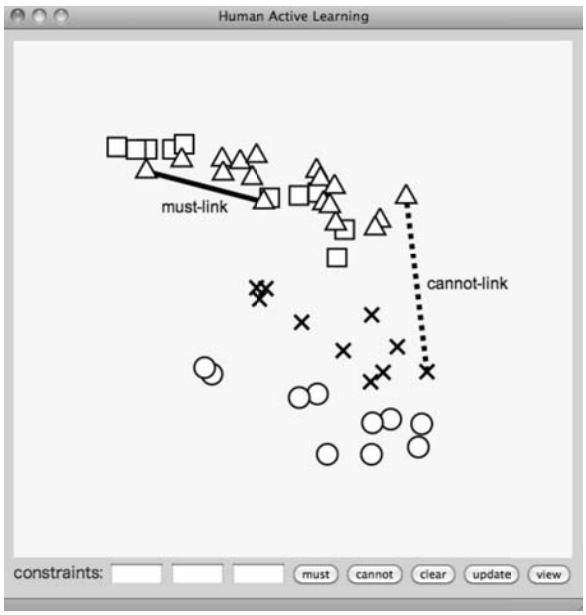
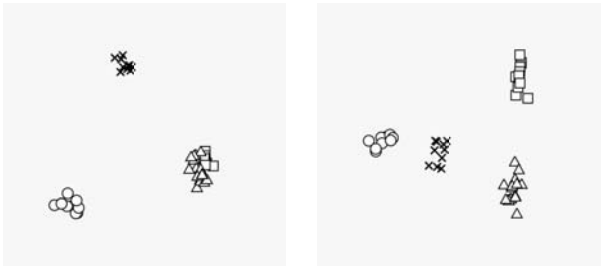


Figure 1. Graphical User Interface



(a) Combination No.1 (b) Combination No.2

Figure 2. Examples of Data Arrangement (two combinations of two axises)

that realizes interactive constraint assignment through the interactive clustering process.

In the following sections, we first explain an overview of our tool in Section 2. Then, we describe the two main functions of the tool - display arrangement by multi-dimensional scaling and incremental distance metric learning, in Sections 3 and 4. Section 5 shows the experimental results where we compare the performance of a heuristic sampling method found by our tool with random sampling and a well-known active learning technique. Section 6 analyzes the data arrangement through the interactions. Finally, we conclude our work in Section 7.

II. System Overview

In this section, we explain the process of interactive constrained clustering when using our proposed tool. Figure 1 shows GUI (Graphical User Interface) of the tool, which consists of some buttons and a 2-D display area to visualize data distribution. Each data is represented by a circle, triangle, rectangle or cross in the 2-D display area. Users can interactively select additional constraints and reflect them to update the clustering result through the GUI. We briefly describe the interaction process between a user and our tool in the following.

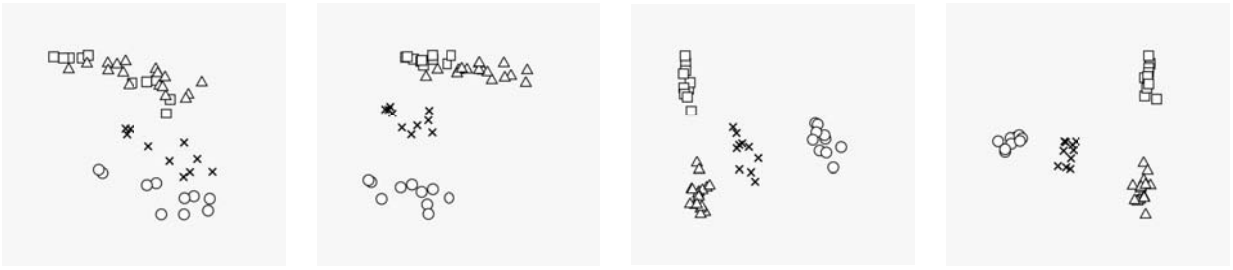
1) A user loads a dataset to be clustered to our tool. Each data must be represented by a feature vector with a pre-defined format. The tool calculates the initial distance matrix from the feature vectors.
2) The tool runs modules of clustering (k-means) and multi-dimensional scaling (MDS) [7] modules to get the temporal clustering result and coordinates of the dataset to display on the GUI. We explain the details behind MDS in the next section. Then, the tool displays data on the GUI according to the 2-D coordinates calculated by the MDS and temporal clustering result.
3) If a user is not satisfied with the clustering results, he/she can add constraints. The tool updates the distance matrix according to additional constraints. We describe the details of this update procedure in Section 4.
4) Repeat steps 2 and 3 until the user is satisfied with the clustering results.

Users can select a pair of data to assign a constraint by clicking the data figures. After selecting the data, users can assign a constraint to it by clicking the “must” or “cannot” buttons. Figure 1 shows examples of must-link (two triangles) and cannot-link (triangle and cross). Then, they update the distance matrix and re-clustering by clicking the “update” button. We use a k-means algorithm for a clustering process.

Most researches in constrained clustering use labeled datasets in their experiments and prepare constraints at random. However, it is clear that random selection is quite wasteful when humans must determine the labels of the constraints. Our tool helps to reduce the labeling cost. In addition, we use selection bias for the constraints because we can recognize the proximity relation between data. This functionality may help users to find better selection strategies.

III. Data Arrangement by Multi-Dimensional Scaling

In this section, we describe a method of data arrangement. When we apply clustering to a dataset, we generally use a high-dimensional feature vector to represent the data that cannot be displayed in our 2-D GUI. We need to display proximity relationships that reflects relations in the original space. We use multi-dimensional scaling (MDS) [7] to realize such 2-D visualization in our tool. MDS is a well-known technique that calculates the spatial arrangement from the distance (or similarity) matrix of a dataset. Although other techniques such as the Self-Organizaing Maps (SOM) [8] or the Generative Topographic Mapping (GTM) [9] can also be candidates, we do not use them in this tool because they are both significantly dependent on the initial seed selection. We give a brief introduction to MDS in the following paragraph.



(a) Result at 1 step (b) Result at 5 step (c) Result at 10 step (d) Result at 15 step

Figure 3. Results of Incremental Distance Learning

MDS is based on the eigen-decomposition, which is a factorization technique of a square matrix. Let S be a square matrix and $\mathbf{v}$ is an eigen vector of S.

$$S\mathbf{v} = \lambda\mathbf{v}$$

λ is an eigen value corresponding to $\mathbf{v}$. Then, S can be factorized as

$$S = V\Lambda V^{-1}.$$

where V is the square matrix whose columns are the eigen vectors of S. Since we calculate S from a symmetric distance matrix D, S is also symmetric. Thus, S can be factorized as

$$\begin{aligned} S &= V\Lambda V^T && (1)\\ &= V\Lambda^{1/2}\Lambda^{1/2}V^T. && (2)\\ &= V\Lambda^{1/2}(V\Lambda^{1/2})^T && (3) \end{aligned}$$

Each row of $V\Lambda^{1/2}$ corresponds to the coordinate of each data in MDS. The dimensions of the coordinate is determined by the number k of eigen values and vectors we take. Since the amount of cumulative contribution ratio is an available measure to explain how well the original proximity relationship is preserved, we use the top k largest eigen values and corresponding eigen vectors to keep original distance as much as possible. Although $k = 2$ is enough to diplay data on the 2-D area, it may not be enough to distinguish clusters if the number of clusters is larger. Thus, we calculate $k > 2$ numbers of eigen values and vectors and use arbitrary combinations of two vectors as coordinates to arrange data in 2-D display area. When users cannot distinguish cluster boundaries or proximity relationship, they can view the data arrangement from another side by changing coordiantes.

Figures 2(a) and 2(b) show the examples of such combinations. We used a "soybean-small" dataset from the UCI repository [10] for Figure 2. Two clusters (triangle and rectangle) are overlapping in Figure 2(a), but are separated in Figure 2(b) because the pairs of axes are different.

S can be calculated from D, which we repeatedly updated through the interaction with our tool. Using a centering matrix G_n, S is calculated as

$$S = \frac{1}{2}G_n D G_n^T,$$

$$G_n = I_n - \frac{1}{n}1_n 1_n'.$$

The centering matrix can remove the effect of the original point.

IV. Algorithm for Incremental Learning of Distance Matrix

In this section, we describe the distance learning algorithm adopted in our tool. This algorithm was proposed by Jain et al. [11] and is based on an online learning framework. It repeatedly updates the distance matrix using the constraints given through an interaction process described in Section 2. It requires less computational cost than other constrained clustering techniques that require some optimization procedures. This advantage is very desirable for our tool, which needs quick response to indicate updated results to users.

The algorithm is based on the problem of learning a Mahalanobis distance function. Given n-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, the squared Mahalanobis distance between them is defined as

$$d_A(\mathbf{u},\mathbf{v}) = (\mathbf{u}-\mathbf{v})^T A(\mathbf{u}-\mathbf{v}),$$

where A is initially the unit matrix and thus $d_A(\mathbf{u},\mathbf{v})$ is initially the Euclid distance between the feature vectors. The objective of the learning is to get a semi-definite matrix A that produces desirable $d_A(\mathbf{u},\mathbf{v})$ for the constrained data pairs. Jain et al. considered updating $d_A(\mathbf{u},\mathbf{v})$ incrementally and proposed an online algorithm that receives one constraint at a time [11]. We briefly describe how they introduced the update formula.

Let A_t be the distance matrix that is updated in the t-th step, and $(\mathbf{u}_t, \mathbf{v}_t, y_t)$ be a constrained data pair given at that time. Here, y_t is the target distance that $d_A(\mathbf{u},\mathbf{v})$ must satisfy. If the data pair is a must-link, y_t is 0. If it is a cannot-link, y_t is 1. Jain et al. formalized an online learning problem to solve A_{t+1} by introducing a regularization function $D(A, A_t)$ and a loss function $l(d_A(\mathbf{u}_t,\mathbf{v}_t), y_t)$ like the following.

$$A_{t+1} = \arg\min_{A\succ 0} D(A, A_t) + \eta l(d_A(\mathbf{u}_t,\mathbf{v}_t), y_t)$$

$$D(A, A_t) = tr(AA_t^{-1}) - \log\det(AA_t^{-1}) - d$$

$$l(d_A(\mathbf{u}_t,\mathbf{v}_t), y_t) = (d_A(\mathbf{u}_t,\mathbf{v}_t) - y_t)^2$$

η is a regularization parameter that determines the degree of constraint's influence. In order to analytically derive an updated formula, $d_A(\mathbf{u}_t, \mathbf{v}_t)$ is approximated by $d_{A_t}(\mathbf{u}_t, \mathbf{v}_t)$. Although we omit the details of the introduction process, the update distance matrix can be analytically solved.

$$d_{A_{t+1}}(\mathbf{u}_t, \mathbf{v}_t) = \frac{\eta y_t \hat{y}_t - 1 + \sqrt{(\eta y_t \hat{y}_t - 1)^2 + 4\eta \hat{y}_t^2}}{2\eta \hat{y}_t}$$

$$\hat{y}_t = d_{A_t}(\mathbf{u}_t, \mathbf{v}_t)$$

Our tool can incrementally change the clustering result based on the distance matrix updated by the above formula. Figure 3 shows a series of cluster changes achieved by using this incremental algorithm. The dataset used in Figure 3 is also the "soybean-small" one. Two clusters (circle and cross) are slightly overlapping in Figure 3(a), but are clearly separated in Figure 3(b). The relationship between two clusters (triangle and rectangle) also changes from Figures 3(b) to 3(c). The clusters in Figure 3(d) seem to be more condensed than those in Figure 3(c). We can see the clusters are gradually separated as the distance learning proceeds.

V. Experiments

We have developed a prototype of this tool. With this tool, we investigated whether we can find any useful heuristics. Through many trials of interaction, we found that users tend to give constraints to certain data pairs, which are selected from a large cluster that may be unseparated from the others, and are must-link pairs spread apart as far as possible within the cluster. We recognize it as a kind of heuristics and investigated its performance by comparing it with a random sampling and an uncertainty sampling. The compared methods are summarized as follows.

1) Human sampling: This is a sampling method using human heuristics. It selects a large cluster and the probable must-link data pairs that are greatly separated in the cluster.
2) Random sampling: This method selects must-link data pairs at random.
3) Uncertainty sampling: This is a sampling method that is based on a well-known active learning approach in the machine learning. Original idea is introduced in the classification problem [12], in which the most preferable data to be labeled is the nearest one to the classification boundary. We arrange this idea for selecting data pairs to be constrained. It selects the nearest probable must-link data pair where each member belong to a cluster different from other's one.

We used only must-link pairs in these experiments because cannot-link constraints often bring about a large performance deterioration and thus creates highly unstable results. Since we found that none of the above three methods took advantage of cannot-link in the experiments, we omit the results with cannot-link constraints.

TABLE I.
Summary of Datasets

Name	Data	Class	Dimension
Soybean-small	47	4	35
Iris	150	3	4
ODP	68	4	662

We used three datasets in the experiments, two datasets from the UCI repository [10] and one from the Open Directory Project. From the UCI repository, we used the "Soybean-small" and "Iris" datasets, which are well-known benchmarks for the machine learning algorithm. The dataset from the ODP was more practical and consists of the Web index pages that are listed in the directories in the ODP. We select four top or sub-top directories from the ODP. They were "Agriculture", "Astronomy", "Math" and "Linguistics". Each directory has a certain number of registered URLs. We retrieve the top pages from these URLs and perform some preprocessing, such as removing the tags and stopwords and stemming. We summarize the characteristics of the three datasets in Table I, where the number of data, class and dimensions of each dataset are listed.

The evaluation measure is normalized mutual information (NMI). The NMI is calculated as follows.

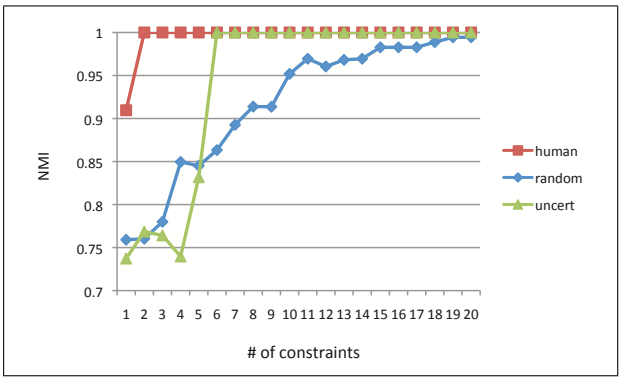
$$\text{NMI}(C, T) = \frac{I(C, T)}{\sqrt{H(C)H(T)}},$$

where C is a set of clusters returned by the k-means algorithm, and T is a set of true clusters. $I(C, T)$ is the mutual information between C and T, and $H(C)$ and $H(T)$ are the entropies.

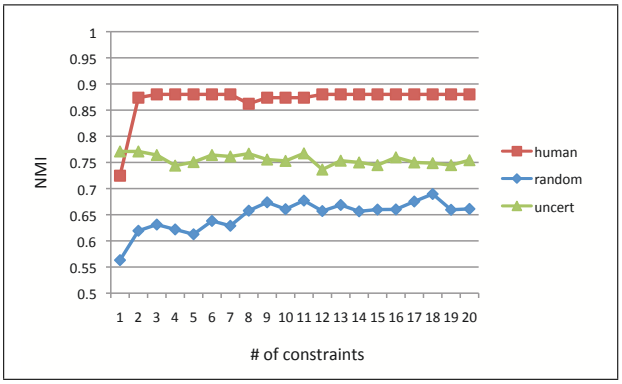
We simulated the interaction process described in Section 2. We repeated 20 steps in each interaction, thus the constraints were increased up to 20. We tested 10 interactions for a dataset and ran k-means algorithm 10 times with randomized seeds in each interaction steps. Thus, the value of NMI was averaged over 10 x 10 trials for a certain number of constraints.

Figures 4(a) and 4(b) shows the results from the Soybean-small and Iris datasets, respectively. In both graphs, human sampling (the label is "human" in the following graphs) thoroughly outperforms other methods (labels are "random" for random sampling and "uncert" for uncertainty sampling, respectively), the effect was especially prominent in the early stage of the interaction. Although both human and uncert reaches the best score, human achieved a few steps earlier. On the other hand, Figures 5(a) and (b) shows the results for the ODP dataset. In this dataset, human and random are comparable in the early stages of interaction, and the human sampling gradually outperforms at around 10 steps passed. We changed the learning rate η to promote the performance of the first few steps. However, there was no effective improvement. The performance of uncert remains comparable or low to random.

These results indicate that our tool can help users to select effective constraints. Moreover, the interactive

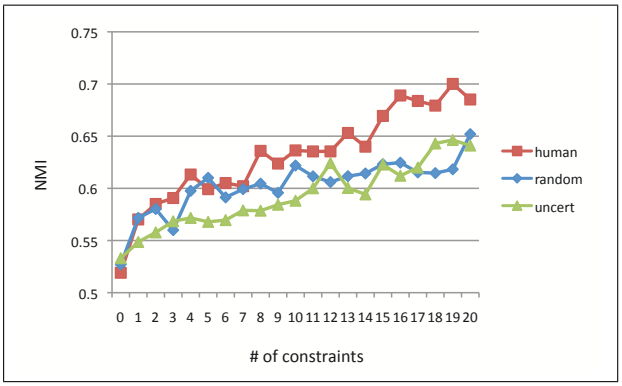


(a) Soybean-small ($\eta = 0.5$)

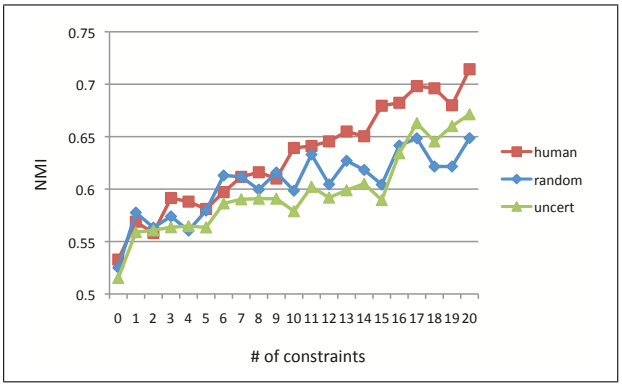


(b) Iris ($\eta = 0.5$)

Figure 4. Results of UCI repository dataset



(a) $\eta = 0.5$



(b) $\eta = 0.9$

Figure 5. Results of Open Directory Project dataset

tool may help embody the user's unconscious heuristic approach by objectively watching interaction. This type of research is related to "human active learning"[13]. Different from their experiments, our interest exists in the "human sampling", where humans only selects the training examples and the learning itself is done by a machine. We consider this is more important for put machine learning to more practical use.

VI. Discussion

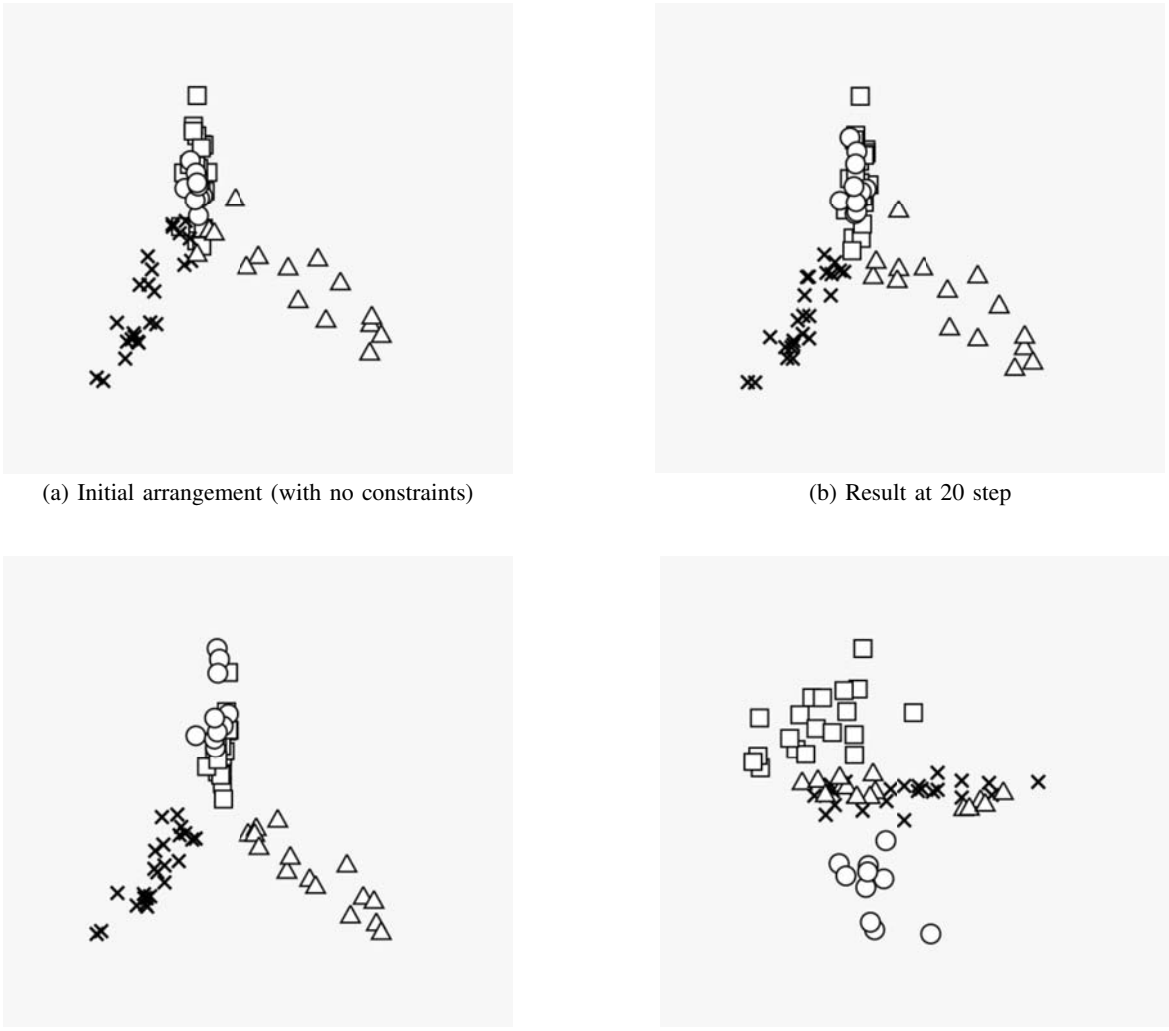
In this section, we analyze how our heuristic sampling changes the data arrangement through the interaction. Figures 6(a)-(d) shows the data arrangements of the ODP datasets. This dataset has 4 classes. For better understanding, data are represented by four types of figures (circle, triangle, rectangle and cross) based on the true class. Figure 6(a) shows the initial data arrangement with a certain axis combination. The difficulty of this dataset lies in the part of intersection of all classes. Thus, one of good strategies is to separate data of this part. Our heuristic first selects one data from the part of the intersection, then searches the other data that is the farthest from and belongs to the same class as the first one, finally make them a must-link pair. Based on this strategy, the part of intersection is gradually separated from each other as shown in Figure 6(b). The performance is improved along with this changes. This change is more clearly recognized as learning proceeded. As for the reference, we put Figure 6(c) that shows the data arrangement after 100 steps. Classes are completely separated from each other. Although two classes (circle and rectangle data) are overlapping, they are separated with another axis combinations. Figure 6(d) shows one of such combinations.

The number of data used in the experiments is relatively small as compared with the normal benchmark datasets. Computational cost of this tool depends on the number of data and the dimensions of original feature vectors. Computational time is almost spent on the calculation of distance matrix and its eigen decomposition in MDS. We need any specific technique or special hardwere if we apply our tool to large volume datasets.

VII. Related work

A. Interactive Clustering

Interactive clustering is a promising approach for implementing intelligent Web interactions because it is very powerful for the interactive visualization, data mining, and data analysis of the Web [14][15], instead of using the conventional clustering methods[16]. Interactive clustering consists of two main techniques: active learning



(a) Initial arrangement (with no constraints)

(b) Result at 20 step

(c) Result at 100 step

(d) Result at 100 step (with another axis combination)

Figure 6. Data Arrangement of ODP after Distance Learning

and constrained clustering. The active learning selects effective data that are judged by the user and used as constraints [12]. The constrained clustering categorizes the data under such constraints judged by the user. Both pursue the common requirement of dealing with the constraints given by the user [17].

desJardins et al. proposed Interactive Visual Clustering (IVC) [18], a novel method that allows a user to interactively explore relational datasets interactively in order to produce a clustering that satisfies their objectives. IVC combines a spring-embedded graph layout with the user interaction and constrained clustering. The experimental results from several synthetic and real-world datasets show that IVC yields a better clustering performance than the alternative methods. This IVC is closely related to our study, however their purpose does not include the acquisition of human heuristics for active learning. In contrast with it, our purpose is to develop an interactive system in which a user can easily acquire his/her heuristics for the active learning of the system.

B. Clustering and visualization on the Web

Clustering also has been used for the Web in the various ways systems[19], [20], [21], [6], [22], [14], [23], [24], [25]. In addition, various visualization techniques for clustering Web pages have been developed so far[26], [27], [28]. Most of them cluster the searched Web page. Ding et al. proposed a HiCluster system [23]. Images searched for by using an image search engine contain multiple topics on the semantic level. Furthermore, even semantically consistent images have diverse appearances on the visual level. It is important to organize such results into semantically and visually consistent clusters for adequately facilitating the users' navigation. To cope with this problem, they developed HiCluster, an effective method to organize image search results, which employs both textual and visual analysis. They applied a K-lines-based clustering algorithm, a Bregman Bubble Clustering algorithm, and a novel helpful UI based on the hierarchical clustering structure for the system.

Ramage et al. proposed a method to employee tagged information for clustering in the Web [19]. They explored

K-means clustering in an extended vector space model including tags as well as page text and a clustering algorithm that jointly models both the text and tags. They also found that the naive inclusion of tagging data improved the cluster quality, however a more principled inclusion can substantially improve the quality.

Although other various clustering for Web systems have been developed [6], a majority of them focused on the clustering results of the search engine. We agree that such clustering is important for Web applications, especially Web searches. However, clustering Web pages is still significant for data mining with a huge amount of Web data. We proposed a promising interactive framework to deal with such noisy Web pages and showed its feasibility.

C. Constrained Clustering

Constrained clustering is also an important function for our interactive clustering. Semi-supervised learning creating classifiers or clusters from limited supervised information and a lot of unlabeled data has been vigorously researched [29]. In this framework, constrained clustering is used as a learning problem. In the typical setting for constrained clustering, several pairwise data are given as either must-link or cannot-link constraints.

One simple use of these constraints is to build a procedure into clustering algorithms to check whether clusters break these constraints. COP-K-means proposed by Wagstaff [1] is one of such representative methods. While this approach is simple, it is too difficult to create consistent clusters with these constraints as the must/cannot-link pairs increases.

Meanwhile, another approach uses constraints to modify the similarities between data as the similarities of must-link pairs must be large and the similarities of cannot-link pairs must be small. There are several methods to realize this approach [2], [30], [31], [3], [4], [11].

Metric learning algorithms are able to produce useful distance functions for constrained clustering in various domains. Although most of such work assumes that sufficient constraints are given all at once, the constraints are only incrementally available in practical applications. Thus, Jain et al. proposed online metric learning [11]. They presented a new online metric learning algorithm that updates a learned Mahalanobis metric and conducts empirical evaluations to compare the proposed method against the existing online metric learning algorithms.

In order to implement our interactive system, we needed an online constrained clustering algorithm because the constraints are given one by one from a user. Therefore, we applied the online metric learning algorithm [11] by Jain et al. because it can deal with the constraints one by one and it outperforms other conventional online constrained clustering methods.

D. Open Directory Project as Source of a Dataset

The Open Directory Project (ODP)[1] is a large human-edited directory of the Web and it is constructed and constantly maintained by a global community of volunteer editors. ODP provides a large directory that covers various fields like the arts, business, computers, games, health, home, kids, recreation, and so on. We can easily access and utilize on a large number of Web pages through the well-structured directory.

Various studies on the Web have utilized ODP. One of ODP's typical usages is to use it as a well-classified dataset of Web pages [32], [33]. One of our main purposes in this work is to apply the proposed interactive clustering to Web page clustering. Thus we built a dataset of Web pages by using the structural information in ODP.

[1] http://www.dmoz.org/

VIII. Conclusion

This paper presented an interactive tool for constrained clustering that provides some basic functions such as the display of a 2-D visual arrangement of a dataset, constraint assignment through mouse manipulation, incremental learning of a distance matrix, and clustering by k-means. These functions help users intervene in the process of constrained clustering, and finally create a satisfactory clustering result with less user cognitive load than that for a clustering process under randomly selected constraints. In addition, the selection bias of the constraints may help users find better selection strategies. We consider our proposed GUI is a promising approach for large-scaled applications like Web clustering.

The tool described in this paper is still in the development phase. We are planning to provide more sophisticated functions. For example, displaying data information is a very important function because users determine the labels of the constraints based on the information. However, a methods for displaying them depend on their data type. We need to implement different methods when displaying images and document data. We are also considering implementing an active learning function that is important but rarely explored in constrained clustering, especially interactive constrained clustering. We think this active learning function may help users, or users may notice the drawbacks of the active learning algorithm. We are currently planning to conduct larger scale user studies in Web clustering to evaluate the advantages of our proposed GUI.

References

[1] K. Wagstaff and S. Roger, "Constrained k-means clustering with background knowledge," in *In Proceedings of the 18th International Conference on Machine Learning*, 2001, pp. 577–0584.

[2] 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 *In Proceedings of the 19th International Conference on Machine Learning*, 2002, pp. 307–314.

[3] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 361–368.

[4] Z. Li, J. Liu, and X. Tang, "Pairwise constraint propagation by semidefinite programming for semi-supervised classification," in *ICML '08: Proceedings of the 25th international conference on Machine learning*. New York, NY, USA: ACM, 2008, pp. 576–583.

[5] S. Basu, M. Bilenko, and R. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 59–68.

[6] C. Carpineto, S. Osiński, G. Romano, and D. Weiss, "A survey of web clustering engines," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–38, 2009.

[7] I. Borg and P. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Verlag, 1997.

[8] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 2002.

[9] C. Bishop, M. Svensén, and C. Williams, "GTM: The generative topographic mapping," *Neural computation*, vol. 10, no. 1, pp. 215–234, 1998.

[10] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.

[11] P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," in *In Proceedings of Twenty-Second Annual Conference on Neural Information Processing Systems*, 2008, pp. 761–768.

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

[13] R. Castro, C. Kalish, R. Nowak, R. Qian, T. Rogers, and X. Zhu, "Human active learning," *Advances in Neural Information Processing Systems (NIPS)*, vol. 22, 2008.

[14] W. Wu, C. Yu, A. Doan, and W. Meng, "An interactive clustering-based approach to integrating source query interfaces on the deep web," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004, pp. 95–106.

[15] K. Bade and A. Nurnberger, "Personalized hierarchical clustering," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 181–187.

[16] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[17] S. Basu, I. Davidson, and K. Wagstaff, Eds., *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall, 2008.

[18] M. desJardins, J. MacGlashan, and J. Ferraioli, "Interactive visual clustering," in *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2007, pp. 361–364.

[19] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," in *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2009, pp. 54–63.

[20] P. Ferragina and A. Gulli, "A personalized search engine based on web-snippet hierarchical clustering," in *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005, pp. 801–810.

[21] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 1–31, 2009.

[22] F. Geraci, M. Pellegrini, P. Pisati, and F. Sebastiani, "A scalable algorithm for high-quality clustering of web snippets," in *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2006, pp. 1058–1062.

[23] H. Ding, J. Liu, and H. Lu, "Hierarchical clustering-based navigation of image search results," in *MM '08: Proceeding of the 16th ACM international conference on Multimedia*. New York, NY, USA: ACM, 2008, pp. 741–744.

[24] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of www image search results using visual, textual and link information," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2004, pp. 952–959.

[25] J. S. Whissell, C. L. Clarke, and A. Ashkan, "Clustering web queries," in *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2009, pp. 899–908.

[26] Z.-W. Li, X. Xie, H. Liu, X. Tang, M. Li, and W.-Y. Ma, "Intuitive and effective interfaces for www image search engines," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2004, pp. 748–749.

[27] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, and W.-Y. Ma, "Igroup: web image search results clustering," in *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2006, pp. 377–384.

[28] O. Gerstel, S. Kutten, E. S. Laber, R. Matichin, D. Peleg, A. A. Pessoa, and C. Souza, "Reducing human interactions in web directory searches," *ACM Transactions on Information Systems*, vol. 25, no. 4, p. 20, 2007.

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

[30] S. Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *In Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 94–101.

[31] W. Tang, H. Xiong, S. Zhong, and J. Wu, "Enhancing semi-supervised clustering: a feature projection perspective," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2007, pp. 707–716.

[32] S. Osinski and D. Weiss, "Conceptual clustering using lingo algorithm: Evaluation on open directory project data," in *In Proceedings of Intelligent Information Processing and Web Mining*, 2004, pp. 369–377.

[33] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter, "Using odp metadata to personalize search," in *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2005, pp. 178–185.

Masayuki Okabe is a research associate at Toyohashi University of Technology. He received B.S. (1996) degree from Soka University and M.S. (1998) and the Ph.D. (2001) degrees from Tokyo Institute of Technology. His research interests include information retrieval, machine learning and data mining. He is a member of The Japanese Society for Artificial Intelligence.

Seiji Yamada is a professor at the National Institute of Informatics. Previously he worked at Tokyo Institute of Technology. He received B.S. (1984), M.S. (1986) and the Ph.D. (1989) degrees in artificial intelligence from Osaka University. His research interests are in the design of intelligent interaction including Human-Agent Interaction, intelligent Web interaction and interactive machine learning. He is a member of IEEE, AAAI, ACM, JSAI, IPSJ and HIS.