

# Constrained Agglomerative Hierarchical Clustering Algorithms with Penalties

Sadaaki Miyamoto

Department of Risk Engineering

University of Tsukuba

Ibaraki 305-8573, Japan

Email: miyamoto@risk.tsukuba.ac.jp

Akihisa Terami

Master's Program in Risk Engineering

University of Tsukuba

Ibaraki 305-8573, Japan

Email: e0611387@edu.esys.tsukuba.ac.jp

**Abstract**—Semi-supervised clustering with constraints has widely been studied, but there are few studies on constrained agglomerative hierarchical algorithms. We have shown modified kernel algorithms of agglomerative hierarchical clustering, but there is a drawback that the modified kernels are not positive-definite in general. In this paper we consider another idea of agglomerative hierarchical algorithms with pairwise constraints. That is, merging of clusters is with penalties. The centroid method and the Ward method with and without a kernel are considered. Typical numerical examples show effectiveness of the proposed algorithms in generating clusters with nonlinear cluster boundaries. We also compare the results with those by COP  $K$ -means, showing that the proposed algorithms outperform the COP  $K$ -means.

**Keywords**—semi-supervised clustering; pairwise constraints; agglomerative hierarchical clustering.

## I. INTRODUCTION

As cluster analysis is becoming a standard tool for data mining, various techniques have been developed and at the same time fundamental problems to be solved are addressed; three of them are as follows:

- 1) how traditional methods of agglomerative hierarchical clustering [6], [9], [10] should be used;
- 2) how clusters with nonlinear boundaries [11] should be generated;
- 3) how to semi-supervised clustering should be handled.

These three problems have sound reasons why they are important. First, agglomerative hierarchical clustering is frequently used for application to many contemporary problems including bioinformatics. Second, the theory of positive-definite kernels [1], [13] has been found to be useful to have nonlinearity in classification and clustering. Third, there are many studies in semi-supervised clustering using different ideas [4]. As to the last problem, various researches have been revealed that clustering with pairwise constraint is useful as a technique to handle semi-supervised classification [15], [2], [3], [8].

This paper concerns all these three problems and intends to give partial solutions to them. First, it proposes a constrained centroid method and a constrained Ward method of agglomerative hierarchical clustering. In contrast to our former paper [12] which uses a modified kernel function, the method herein does not use any kernel function and yet generates clusters with nonlinear boundaries, using a simple method of penalties

in merging clusters. This paper also shows that the standard method of COP  $K$ -means [15] with pairwise constraint cannot generate clusters with nonlinear boundaries without a kernel function, while the present method can do this.

There are well-known typical examples with nonlinear boundaries that we can handle by the present method. We show the effectiveness of the proposed method and moreover compare performances of the proposed method and the COP  $K$ -means which shows the usefulness of the proposed method.

## II. AGGLOMERATIVE CLUSTERING WITH PENALTIES

Let the set of objects for clustering be  $X = \{x_1, \dots, x_N\}$ . We assume that object  $x_k$  ( $k = 1, \dots, N$ ) is a point of a Euclidean space  $\mathbf{R}^p$  with the Euclidean norm  $\|\cdot\|$ .

Generally a hard cluster denoted by  $G_i$  is a subset of  $X$ . The family of clusters is denoted by  $\mathcal{G} = \{G_1, G_2, \dots, G_K\}$ , where the clusters form a crisp partition of  $X$ :

$$\bigcup_{i=1}^K G_i = X, \quad G_i \cap G_j = \emptyset \quad (i \neq j).$$

Agglomerative hierarchical clustering uses a dissimilarity measure, frequently called a distance, between two clusters  $d(G, G')$  ( $G, G' \in \mathcal{G}$ ) which also is called an inter-cluster dissimilarity. In the following we sometimes write  $d_{cn}(G, G')$  which means that the centroid method is used. Moreover,  $d_{wm}(G, G')$  means that the Ward method is used. The two methods use the following definitions of the inter-cluster dissimilarity:

### Centroid method:

Let the centroid of a cluster  $G$  be

$$M(G) = \frac{1}{|G|} \sum_{x_k \in G} x_k, \quad (1)$$

and let

$$d_{cn}(G, G') = \|M(G) - M(G')\|^2. \quad (2)$$

### Ward method:

Assume  $M(G)$  is the centroid as above, and let

$$E(G) = \sum_{x_i \in G} \|x_i - M(G)\|^2. \quad (3)$$

Define

$$d_{wm}(G, G') = E(G \cup G') - E(G) - E(G'). \quad (4)$$

Although there are other three methods of the single linkage, complete linkage, and average linkage are often used, they are not studied in this paper.

Let us first describe a general procedure of agglomerative hierarchical clustering [9], [10] with a variation that a term  $\mathcal{P}(G, G')$  is attached in the merging step, which is called **AHCP** (Agglomerative Hierarchical Clustering with Penalties) here. The definition of this term is described after the procedure.

#### AHCP Algorithm:

**AHCP1:** Assume that initial clusters are given by

$$\mathcal{G} = \{\hat{G}_1, \hat{G}_2, \dots, \hat{G}_{N_0}\}.$$

Set  $K = N_0$ .

( $K$  is the number of clusters and  $N_0$  is the initial number of clusters)

$$G_i = \hat{G}_i \quad (i = 1, \dots, K).$$

Calculate  $d(G, G')$  for all pairs  $G, G' \in \mathcal{G}$ .

**AHCP2:** Search the pair of minimum dissimilarity:

$$(G_p, G_q) = \arg \min_{G, G' \in \mathcal{G}} \{d(G, G') + \mathcal{P}(G, G')\}. \quad (5)$$

Merge:  $G_r = G_p \cup G_q$ .

Add  $G_r$  to  $\mathcal{G}$  and delete  $G_p, G_q$  from  $\mathcal{G}$ .

$K = K - 1$ .

if  $K = 1$  then stop and output the dendrogram.

**AHCP3:** Update dissimilarity  $d(G_r, G'')$  for all  $G'' \in \mathcal{G}$ .

Go to **AHCP2**.

#### End AHCP.

Assume that  $K_0$  is given and it is necessary to have  $K_0$  clusters, then we stop **AHCP** algorithm when  $K_0$  clusters are obtained in **AHCP2**.

#### A. Pairwise constraints and penalties

A set  $ML = \{(x_i, x_j)\} \subset X \times X$  consists of *must-link* pairs so that  $x_i$  and  $x_j$  should be in a same cluster, while another set  $CL = \{(x_k, x_l)\} \subset X \times X$  consists of *cannot-link* pairs so that  $x_i$  and  $x_j$  should be in different clusters.  $ML$  and  $CL$  are assumed to be symmetric in the sense that if  $(x_i, x_j) \in ML$  then  $(x_j, x_i) \in ML$ , and if  $(x_k, x_l) \in CL$  then  $(x_l, x_k) \in CL$ .

Note that  $ML$  is regarded as an undirected graph in which nodes are objects appeared in  $ML$ , and an undirected edge is  $(x_i, x_j) \in ML$ . Accordingly,  $V(ML)$  is the set of objects in graph  $ML$ . The connected components of graph  $ML$  is denoted by  $ML_1, \dots, ML_\ell$ . Moreover the constraints are assumed to be consistent in the sense that if  $(x_k, x_l) \in CL$ ,  $x_k$  and  $x_l$  are in different components of graph  $ML$ .

The must-link constraints are realized in **AHCP** algorithm by putting the initial clusters  $\mathcal{G} = \{\hat{G}_1, \hat{G}_2, \dots, \hat{G}_{N_0}\}$  to be the connected components  $ML_1, \dots, ML_\ell$ .

In contrast, the following penalty term realizes the cannot-link constraints in **AHCP**:

$$\mathcal{P}(G, G') = W_{CL} \sum_{x \in G, x' \in G', (x, x') \in CL} \quad (6)$$

where  $W_{CL}$  is a positive constant. If  $W_{CL}$  is sufficiently large, the cannot-link constraints will never be broken.

*Note:* We handle  $ML$  to be hard constraints, while  $CL$  is treated to be *soft* constraints in (6). It is, however, straightforward to treat  $ML$  as soft constraints as well. For this purpose we modify (6) as follows:

$$\mathcal{P}(G, G') = W_{CL} \sum_{x \in G, x' \in G', (x, x') \in CL} - W_{ML} \sum_{x \in G, x' \in G', (x, x') \in ML} \quad (7)$$

Note also that the  $CL$  constraints become hard when  $W_{CL}$  is sufficiently large, which we assume later for the examples herein.

#### B. Positive-definite kernels in agglomerative clustering

It has been known that a high-dimensional mapping with a kernel function which is used in support vector machines [14], [13], [1] is also useful in agglomerative hierarchical clustering [5]. That is, regardless whether a high-dimensional mapping is used or not, the updating formulas for merging clusters remain the same in the centroid method and the Ward method [5]:

$$\begin{aligned} d_{cm}(G_r, G'') &= \frac{|G_p|}{|G_p| + |G_q|} d_{cm}(G_p, G'') \\ &+ \frac{|G_q|}{|G_p| + |G_q|} d_{cm}(G_q, G'') \\ &- \frac{|G_p||G_q|}{(|G_p| + |G_q|)^2} d_{cm}(G_p, G_q), \end{aligned} \quad (8)$$

$$\begin{aligned} d_{wm}(G_r, G'') &= \frac{1}{|G_p| + |G_q| + |G''|} \left[ \right. \\ &(|G_p| + |G''|) d_{wm}(G_p, G'') \\ &+ (|G_q| + |G''|) d_{wm}(G_q, G'') \\ &\left. - |G''| d_{wm}(G_p, G_q) \right]. \end{aligned} \quad (9)$$

except that the initial dissimilarities are different:

$$\begin{aligned} d_{cm}(x, y) &= \|x - y\|^2 = K(x, x) + K(y, y) - 2K(x, y), \\ d_{wm}(x, y) &= \frac{1}{2} \|x - y\|^2 = \frac{1}{2} \{K(x, x) + K(y, y) - 2K(x, y)\}. \end{aligned}$$

for two objects  $x, y \in X$  from which  $d_{cm}(\hat{G}, \hat{G}')$  and  $d_{wm}(\hat{G}, \hat{G}')$  are easily calculated.

We compare the results from the centroid method and the Ward method with and without the Gaussian kernel:

$$K(x, y) = \exp(-const. \|x - y\|^2).$$

### III. NUMERICAL EXAMPLES

Three illustrative examples are shown, in which cluster boundaries are nonlinear. First set of data is shown in Fig. 1 which has two obvious clusters of “double circles”; the outer circle has 100 objects while the inner has 50. Second data set in Fig. 2 has a ball inside a shape like a crescent, which is called “a moon and a ball.” Third set of data in Fig. 3 is called “two moons” which also cannot be separated by a line. Each “moon” has 150 objects.

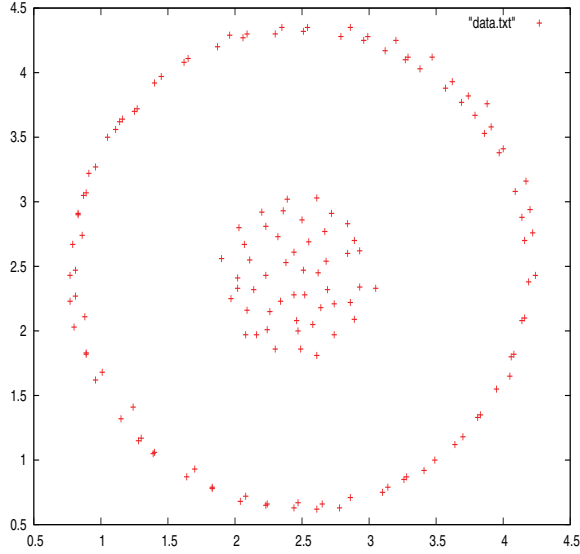


Fig. 1. First data set: double circles

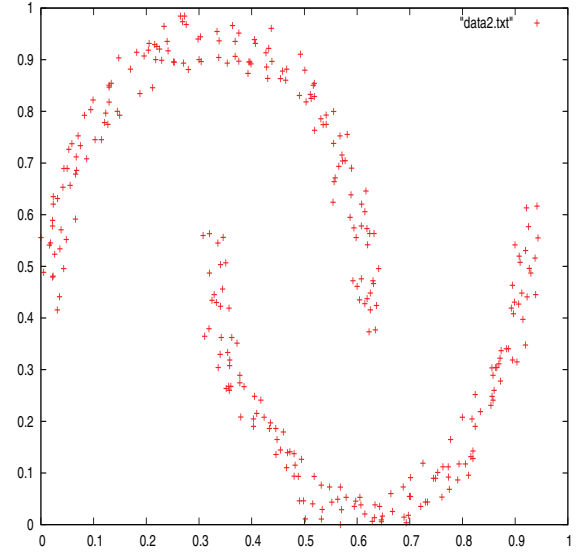


Fig. 3. Third data set: two moons

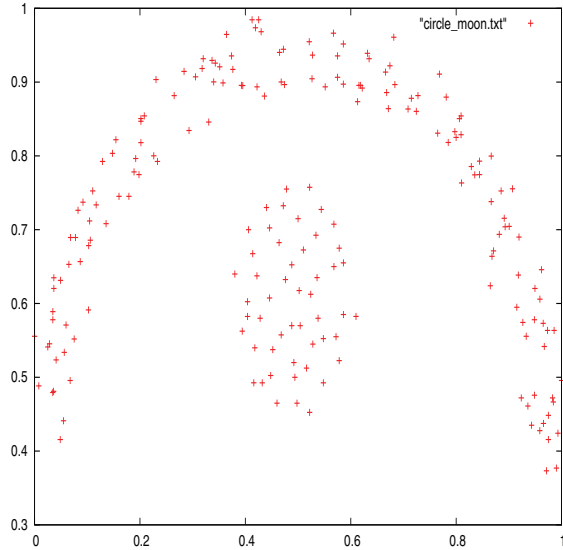


Fig. 2. Second data set: A moon and a ball

We generated  $ML$  and  $CL$  randomly, with the obvious restriction that  $ML$  are in the same (correct) cluster and  $CL$  connects two objects in different clusters. The penalty  $W_{CL}$  was sufficiently large so that  $CL$  pair did not be in a same cluster. Thus  $CL$  was handled as a hard constraint.

The numbers of constraints in  $ML$  and  $CL$  were changed in the numerical experiment and the ratios of correct results were evaluated as follows.

The Rand index which is one of the standard evaluation criteria was used:

$$Rand(a, b) = \frac{a + b}{N(N-1)/2}, \quad (10)$$

where  $a$  is the number of pairs that has to be in the same

cluster and actually in the same cluster in the result, and  $b$  is the number of pairs that has to be in different clusters and actually in different clusters in the results. In short,  $Rand(a, b)$  is the ratio of correct pairs in the whole number of pairs. Hence if the result is perfect,  $Rand(a, b) = 1$ .

The proposed method was compared with the COP  $K$ -means [15] with  $K = 2$  in both examples.

We fixed the number of constraints and then generated the constraints randomly 1000 times, then generated clusters by the proposed method and the COP  $K$ -means. Then the averages of the rand index were calculated.

The number of constraints were then changed from 0 to 50 shown on the horizontal axis in Figs. 4–9.

In the following six kinds of results are compared; we did not mix  $ML$  and  $CL$  but included  $ML$  or  $CL$  alone:

- 1) results of the centroid method with  $ML$  (and no  $CL$ ), shown by red line with + in Figs. 4–9;
- 2) results of the centroid method with  $CL$  (and no  $ML$ ) shown by green line with ×;
- 3) results of the Ward method with  $ML$  (and no  $CL$ ) shown by blue line with \*;
- 4) results of the Ward method with  $CL$  (and no  $ML$ ) shown by pink line with □;
- 5) results of the COP  $K$ -means with  $ML$  (and no  $CL$ ) shown by light blue line with ■;
- 6) results of the COP  $K$ -means with  $CL$  (and no  $ML$ ) shown by black line with ●.

Figures 4 and 5 respectively show the results without a kernel and those with the Gaussian kernel applied to the data set of the double circles. The all results from the COP  $K$ -means was without a kernel. Note that it has been known that  $K$ -means with the Gaussian kernel can separate the two clusters of the two circles.

It is observed that the centroid and the Ward methods with

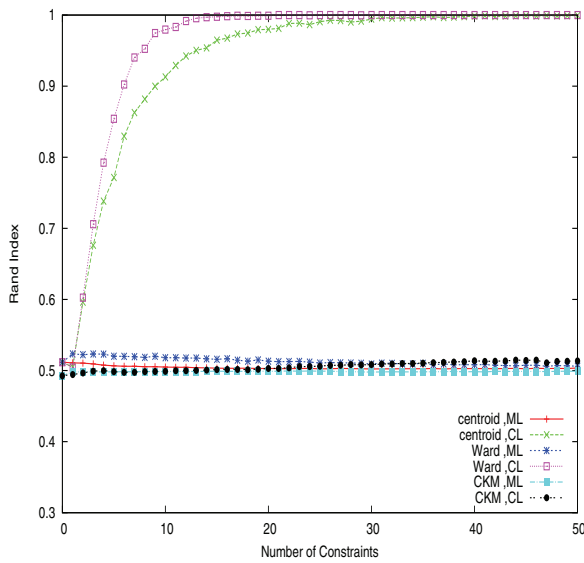


Fig. 4. The ratios of  $Rand(a, b)$  by the proposed method without a kernel and the COP  $K$ -means applied to the double circle data set

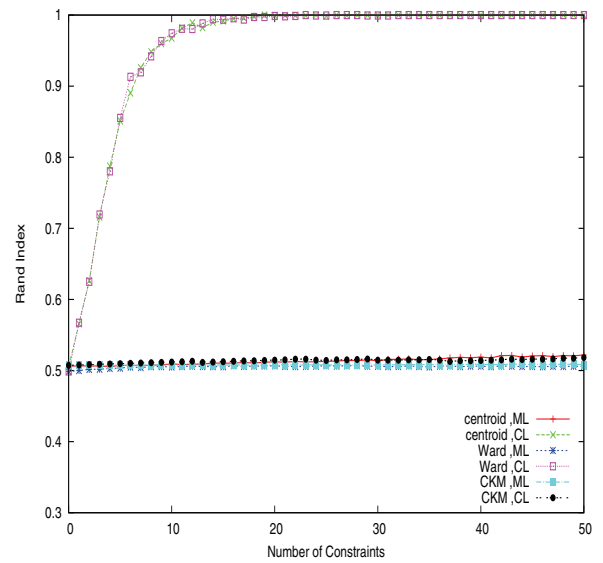


Fig. 6. The ratios of  $Rand(a, b)$  by the proposed method without a kernel and the COP  $K$ -means applied to the moon and ball data set

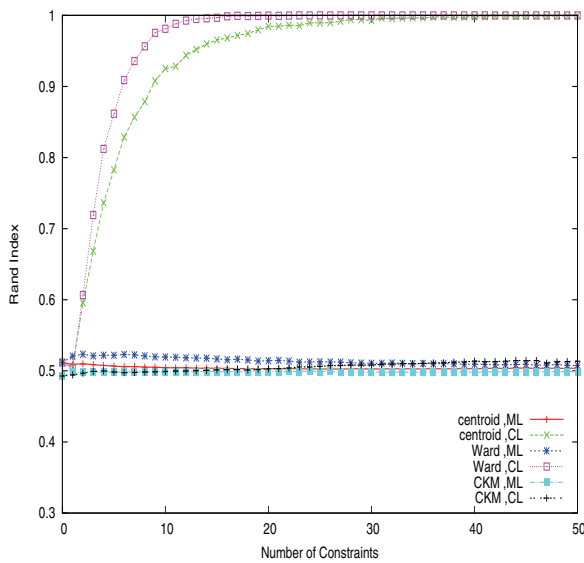


Fig. 5. The ratios of  $Rand(a, b)$  by the proposed method with the Gaussian kernel ( $const. = 0.01$ ) and the COP  $K$ -means applied to the double circle data set

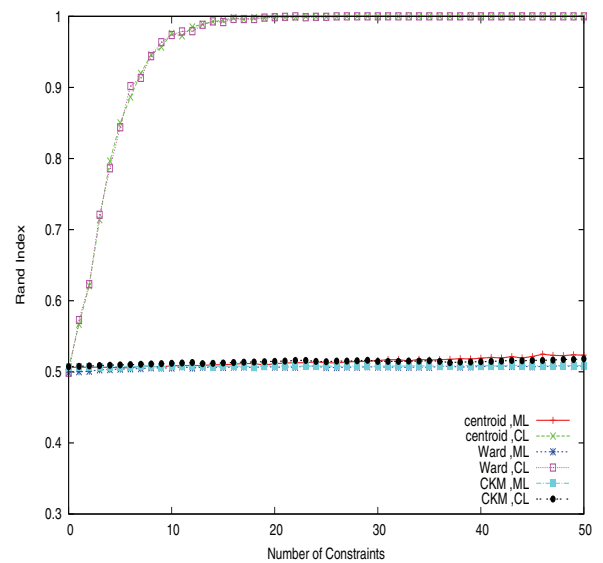


Fig. 7. The ratios of  $Rand(a, b)$  by the proposed method with the Gaussian kernel ( $const. = 0.01$ ) and the COP  $K$ -means applied to the moon and ball data set

$CL$  better separated the two clusters as the number of the cannot-link constraints increase, while the other methods could not separate these two.

Figures 6 and 7 respectively show the results without a kernel and those with the Gaussian kernel applied to the data set of the “moon and ball.” The same tendency as the first example is observed: the centroid and the Ward methods with  $CL$  better separated the two clusters as the number of the cannot-link constraints increase, whereas the other methods could not separate the moon and the ball.

Figures 8 and 9 respectively show the results without a

kernel and those with the Gaussian kernel applied to the data set of the two moons. The COP  $K$ -means was used again without a kernel function. Note that in this case it is generally difficult to separate the two clusters using  $K$ -means, even with the Gaussian kernel.

It is observed again that the centroid and Ward methods with  $CL$  separated the two moons as the number of the cannot-link constraints increase, while the other methods could not separate the two clusters. It is moreover seen that the Gaussian kernel improved the performance in the second example when compared with the centroid and the Ward methods without a

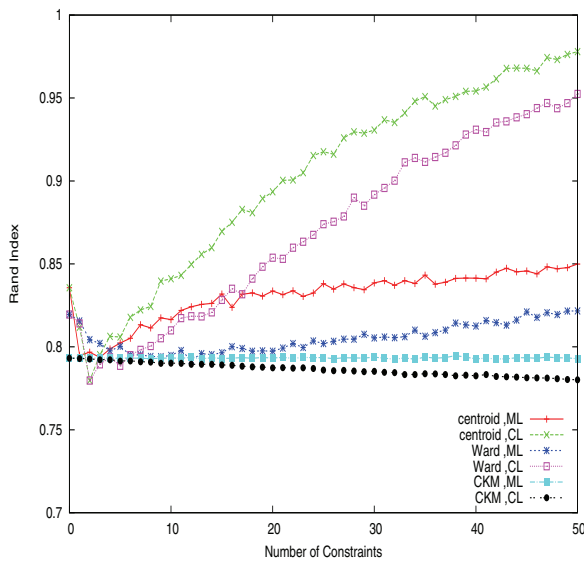


Fig. 8. The ratios of  $Rand(a, b)$  by the proposed method without a kernel and the COP  $K$ -means applied to the two moons data set

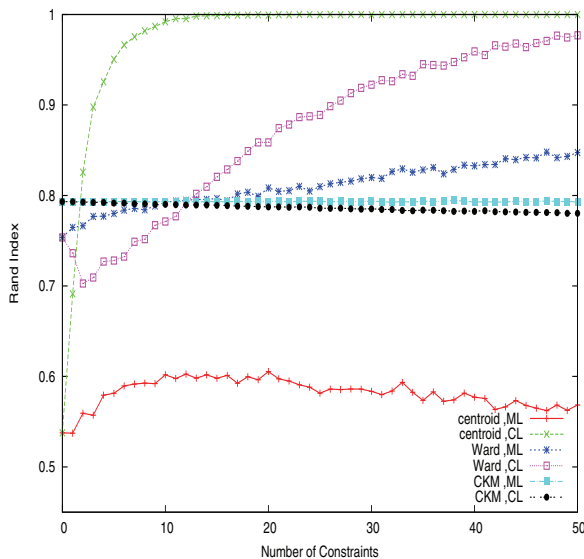


Fig. 9. The ratios of  $Rand(a, b)$  by the proposed method with the Gaussian kernel ( $const. = 10$ ) and the COP  $K$ -means applied to the two moons data set

kernel.

#### A. Discussion

The three illustrative examples show that the proposed method with  $CL$  outperforms other methods. It is also observed that the two clusters in the first and the second examples were separated without the use of a kernel function. This contrasts the general understanding that  $K$ -means type methods including fuzzy  $c$ -means without a kernel cannot separate clusters with nonlinear boundaries. Even if we use the COP  $K$ -means, the cluster boundaries remained the same with the

exception of the pairwise constraints, and hence no perfect separation of the two clusters was observed for the COP  $K$ -means.

In contrast, the third data set of two moons could perfectly be separated with the use of the Gaussian kernel, which shows the effectiveness of a kernel function, as it generally is more difficult to separate the two moons than the two circles.

The three examples moreover show the generalization ability of the  $CL$  constraints especially in the first two examples, since the two clusters were almost perfectly classified when the number of  $CL$  constraints is 10, where the complete cannot links should have 50 constraints.

In contrast, we observed  $ML$  was not so effective as  $CL$ . Why  $ML$  was not so effective in agglomerative algorithms should be further investigated. We also observed that the increase in the number in  $ML$  did not result in the improvement of the rand index. The reason is unknown but similar tendency has also been observed in other studies (e.g. [7]).

#### IV. CONCLUSION

We have proposed the centroid method and the Ward method of agglomerative hierarchical clustering with penalties of the pairwise constraints. The three illustrative examples showed their effectiveness in extracting clusters with nonlinear boundaries. The idea herein is simple and can be implemented without difficulty. Note that agglomerative hierarchical clustering is frequently used in a variety of applications. We thus showed that the proposed method enhances the usefulness of the agglomerative algorithms.

When compared with our former proposal which uses modification of a kernel function [12], the present method is better in the sense that a kernel function is unnecessary and it does not have a drawback of indeterminate kernel which the former method of us has.

Future studies include testing its effectiveness in real examples and dendrogram output showing pairwise penalties. Comparison with fuzzy  $c$ -means with semi-supervision [16] which is not yet an established technique should also be considered.

#### REFERENCES

- [1] S. Akaho, *Kernel Multivariate Analysis*, Iwanami-Shoten, Tokyo, 2008 (in Japanese).
- [2] S. Basu, M. Bilenko, R.J. Mooney, A Probabilistic Framework for Semi-Supervised Clustering, *Proc. of the Tenth ACM SIGKDD (KDD-2004)*, pp.59-68, 2004.
- [3] S. Basu, A. Banerjee, R.J. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, *Proc. of the SIAM International Conference on Data Mining, (SDM-2004)*, pp. 333-344, 2004.
- [4] O. Chapelle, B. Schölkopf, A. Zien, eds., *Semi-Supervised Learning*, MIT Press, 2006.
- [5] Y. Endo, H. Haruyama, T. Okubo, On some hierarchical clustering algorithms using kernel functions, *Proc. of FUZZ-IEEE 2004*, Budapest, Hungary, July 25-29, 2004, CD-ROM Proc. pp. 1-6.
- [6] B.S. Everitt, *Cluster Analysis, 3rd Ed.*, Arnold, London, 1993.
- [7] D. Klein, S.D. Kamvar, C. Manning, From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering, *Proc. of the Intern. Conf. on Machine Learning*, pp.307-314, Sydney, Australia, 2002.
- [8] B. Kulis, S. Basu, I. Dhillon, R. Mooney, Semi-supervised graph clustering: a kernel approach, *Mach. Learn.*, Vol.74, pp.1-22, 2009.

- [9] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Kluwer, Dordrecht, 1990.
- [10] S. Miyamoto, *Introduction to Cluster Analysis: Theory and Applications of Fuzzy Clustering*, Morikita-Shuppan, Tokyo, 1990 (in Japanese).
- [11] S. Miyamoto, H. Ichihashi, K. Honda, *Algorithms for Fuzzy Clustering*, Springer, 2008.
- [12] S. Miyamoto, A. Terami, Semi-Supervised Agglomerative Hierarchical Clustering Algorithms with Pairwise Constraints, *Proc. of WCCI 2010 IEEE World Congress on Computational Intelligence*, July, 18-23, 2010 - CCIB, Barcelona, Spain, pp.2796-2801 (2010).
- [13] B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, 2002.
- [14] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
- [15] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained K-means Clustering with Background Knowledge, *Proc. of the 9th ICML*, pp. 577-584, 2001
- [16] N. Wang, X. Li, X. Luo, Semi-supervised Kernel-based Fuzzy *c*-Means with Pairwise Constraints, *Proc. of WCCI 2008*, pp.1099-1103, 2008.