Genetic-Guided Semi-Supervised Clustering Algorithm with Instance-Level Constraints

Yi Hong
Department of Comp. Sci.
City University of Hong Kong
yihong@cityu.edu.hk

Sam Kwong
Department of Comp. Sci.
City University of Hong Kong
CSSAMK@cityu.edu.hk

Hui Xiong
MSIS Department
Rutgers University
hxiong@rutgers.edu

Qingsheng Ren
Dep. of Comp. Sci. and Eng.
Shanghai Jiao Tong University
ren-qs@cs.sjtu.edu.cn

ABSTRACT

Semi-supervised clustering with instance-level constraints is one of the most active research topics in the areas of pattern recognition, machine learning and data mining. Several recent studies have shown that instance-level constraints can significantly increase accuracies of a variety of clustering algorithms. However, instance-level constraints may split the search space of the optimal clustering solution into pieces, thus significantly compound the difficulty of the search task. This paper explores a genetic approach to solve the problem of semi-supervised clustering with instance-level constraints. In particular, a novel semi-supervised clustering algorithm with instance-level constraints, termed as the hybrid genetic-guided semi-supervised clustering algorithm with instance-level constraints (Cop-HGA), is proposed. Cop-HGA uses a hybrid genetic algorithm to perform the search task of a high quality clustering solution that is able to draw a good balance between predefined clustering criterion and available instance-level background knowledge. The effectiveness of Cop-HGA is confirmed by experimental results on several real data sets with artificial instance-level constraints.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning

General Terms

Algorithms

Keywords

Genetic Algorithms, Semi-Supervised Clustering

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

1. INTRODUCTION

Clustering analysis works to classify a set of unlabeled instances into groups such that instances in the same group are more similar to each other, while they are more different in different groups. In its traditional literature, clustering analysis was considered as an unsupervised method for data analysis, which performs under the condition that no information is available concerning memberships of instances to predefined groups [1]. However, it was known that some background knowledge such as instance-level constraints can be obtained easily in many real-world applications and several recent studies have also shown that these instance-level constraints can significantly increase accuracies of a variety of clustering algorithms. Clustering analysis under the condition that some limited instance-level constraints are incorporated for guiding the clustering of the data was termed as semi-supervised clustering with instance-level constraints, which has become one of the most active research topics in the areas of pattern recognition, machine learning and data mining [2] [3].

Semi-supervised clustering with instance-level constraints has gained some real-world applications such as GPS-based map refinement, person identification from surveilance camera clips and landscape detection from hyperspectral data [2] [3]. However, semi-supervised clustering with instance-level constraints is not exempt from any drawbacks. One disadvantage of semi-supervised clustering with instance-level constraints is that instance-level constraints tend to split the search space of the optimal clustering solution into pieces that compounds the difficulty of the search task. Whereas commonly-used hill-climbing search methods can only guarantee a local optimal clustering solution. The above disadvantage of semi-supervised clustering with instance-level constraints motivates us to adopt genetic algorithms to perform the search task.

This paper explores the genetic approach to solve the problem of semi-supervised clustering with instance-level constraints. In particular, a novel semi-supervised clustering algorithm, termed as the hybrid genetic-guided semi-supervised clustering algorithm with instance-level constraints (Cop-HGA), is proposed. Cop-HGA uses a hybrid genetic algorithm to perform the search task of a high quality clustering solution that is able to draw a good balance between predefined clustering criterion and available instance-level

background knowledge. To the best knowledge of the authors', very few works have been done on using genetic algorithms to solve the problem of semi-supervised clustering with instance-level constraints.

The remainder of this paper is divided into 5 sections. Section 2 briefly introduces the related work for this paper. Section 3 defines the problem of semi-supervised clustering with instance-level constraints. Section 4 goes into details of describing genetic-guided semi-supervised clustering algorithm with instance-level constraints. Our experimental results on several real data sets and their analysis are given in section 5. Section 6 concludes this paper.

2. RELATED WORK

Traditional clustering algorithms are unsupervised under the condition that no information is available concerning memberships of instances to predefined groups. However, if no information about memberships of instances is available, clustering analysis is an ill-posed combinatory optimization problem and no single clustering algorithm is able to achieve high quality clustering solutions for all kinds of data sets. A large number of studies have been concentrated on improving the robustness and stability of clustering algorithms. Among them is semi-supervised clustering algorithms that incorporate some prior background knowledge about memberships of instances into the original framework of traditional unsupervised clustering algorithms. It should be noted that these prior background knowledge can sometimes be obtained naturally from application domains without accessing any human interaction. For example, to segment movies such that all the frames in which the same actor appears are grouped. Due to the continuous nature of most movies, faces extracted from successive frames in roughly the same location can be assumed to come from the same person. Another example is to segment images using clustering algorithms. Two pixels have a high probability to be grouped together if they are spatially connected. Many recent studies have demonstrated that these prior background knowledge can significantly improve accuracies of clustering algorithms [2] [3]. Clustering algorithms incorporate these prior background knowledge in a constrained format, which may come from several different sources such as partial labels, instances relationships and spatial contiguity.

In this paper, we are mainly interested in instance-level constraints, which were known as a more natural representation of prior background knowledge in some scenarios and easier to be collected than accurate labels of instances. For example, in image retrieval systems with user feedback, users are more willing to provide whether a set of retrieved instances are similar or not than to specify labels of instances. Instance-level constraints place restrictions on pairs of instances with regards to their memberships. The concept of instance-level constraints was firstly introduced into the area of clustering analysis in [4], [5] and [3]. To improve the performance of traditional K-means clustering algorithms, two kinds of constraints: Must-link constraints and Cannot-link constraints were proposed and added into the original framework of traditional K-means clustering algorithm. Among these two kinds of instance-level constraints, Must-link ones represent that two instances must be partitioned into the same group, while Cannot-link ones specify that two instances must not be placed into the same group. Other kinds of constraints such as space-level constraints [6] have also been suggested. In this paper, only Must-link and Cannot-link instance-level constraints are considered because of their simplicities and wide applications. There are two widely used approaches for unsupervised clustering algorithms to incorporate instance-level constraints. The first one is to place restrictions on assignments of instances to guarantee that assignments of instances satisfy all given constraints [5] [4]. A variate of this kind of approaches is semi-supervised clustering with penalty that works to penalize clustering solutions according to the degrees in which they violate the given instance-level constraints [7] [8]. The second one is to learn a distance metric from all available instance-level constraints such that instances in the learned distance space are more suitable for the clustering of the data [9]. For more information about semi-supervised clustering algorithms with instance-level constraints, we recommend two PhD thesis [2] [3] and one recent survey about semi-supervised clustering with instance-level constraints [10].

3. PROBLEM DEFINITION

Clustering analysis works to classify a set of unlabeled instances into groups such that instances in the same group are more similar to each other, while they are more different in different groups. Many feasible approaches have been proposed to classify a set of unlabeled instances into groups and most of them belong to the following three categories: data partitioning, hierarchical clustering and model-based clustering. This paper is mainly interested in K-means clustering algorithm, which is one of the most famous data partitioning algorithms. Therefore, in this paper data clustering algorithm works to search for a partition of all instances such that the minimization of the within-cluster variation can be achieved. Let $D = \{x_i, x_2, ..., x_L\}$ represent a set of L instances with m features, x_{ij} denote the j^{th} feature of the instance x_i and K be the number of groups that has been known beforehand. A partition of the data set $C = \{C_1, C_2, ..., C_K\}$ satisfies:

$$C_i \cap C_j = \emptyset \ (i \neq j) \text{ and } \cup_{i=1}^{K} C_i = D \tag{1}$$

Based on the above definitions, the within-cluster variation of the partition C of the data set can be calculated as:

$$s(C) = \sum_{i=1}^{L} \sum_{k=1}^{K} \left[\delta(x_i, C_k) \cdot \sum_{j=1}^{m} (x_{ij} - c_{kj})^2 \right] \tag{2}$$

where

$$c_{kj} = \frac{\sum_{i=1}^{L} \delta(x_i, C_k) \cdot x_{ij}}{\sum_{i=1}^{L} \delta(x_i, C_k)} \tag{3}$$

for $k = 1, ..., K$, $j = 1, ..., m$ and

$$\delta(x_i, C_k) = \begin{cases} 1 & \text{if } x_i \in C_k; \\ 0 & \text{if otherwise;} \end{cases} \tag{4}$$

K-means clustering algorithm employs the following steps to search for a partition C of the data set such that the minimization of the within-cluster variation of instances can be achieved: 1. Cluster centroids are updated based on the labels of all instances. 2. All instances are reassigned and relabeled into its closest cluster centroid. Traditional K-means clustering algorithm is very fast, therefore has been widely applied into the areas of pattern recognition, machine learning and data compression. However, traditional

K-means clustering algorithm can only converge to a local optimal clustering solution. Moreover, the accuracy of K-means clustering algorithm may be not high enough if the distribution of instances is elongated [1].

There are many feasible approaches for improving the robustness of traditional K-means clustering algorithm. Among them is the recently proposed semi-supervised clustering algorithm that incorporates instance-level constraints into the original framework of traditional K-means clustering algorithm [4] [8]. Instance-level constraints represent whether pairs of instances should or should not be classified into the same groups. Two widely used instance-level constraints are Must-link constraints and Cannot-link constraints. Among these two kinds of instance-level constraints, Must-link ones represent that two instances must be partitioned into the same group, while Cannot-link ones specify that two instances must not be placed into the same group. Let Con be the set of instance-level constraints that can be denoted as follows:

$$Con(i,j) = \begin{cases} 1 & \text{if} \quad x_i \text{ and } x_j \text{ is Must-linked;} \\ -1 & \text{if} \quad x_i \text{ and } x_j \text{ is Cannot-linked;} \\ 0 & \text{if} \quad \text{otherwise;} \end{cases} \tag{5}$$

We use the following function to check whether the partition C of the data set satisfies a given instance-level constraint $Con(i,j)$:

$$\delta(C, Con(i,j)) =$$

$$\begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are in the same group, but } Con(i,j) = -1 \\ 1 & \text{if } x_i \text{ and } x_j \text{ are in different groups, but } Con(i,j) = 1 \\ 0 & \text{if otherwise;} \end{cases} \tag{6}$$

Therefore, given a partition $C = \{C_1, C_2, ..., C_K\}$ of the data set, its number of unsatisfied instance-level constraints can be calculated as follows:

$$Vcon(C, Con) = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta(C, Con(i,j)) \tag{7}$$

If the clustering solution C completely satisfies all given constraints Con, then

$$Vcon(C, Con) = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta(C, Con(i,j)) = 0 \tag{8}$$

otherwise,

$$Vcon(C, Con) = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta(C, Con(i,j)) > 0 \tag{9}$$

If all instance-level constraints should be satisfied[1], K-means clustering algorithm with instance-level constraints can be viewed as the following constrained combinatory optimization problem:

$$\min \left\{ \sum_{i=1}^{L} \sum_{k=1}^{K} \left[\delta(x_i, C_k) \cdot \sum_{j=1}^{m} (x_{ij} - c_{kj})^2 \right] \right\} \tag{10}$$

subject to

$$\sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta(C, Con(i,j)) = 0$$

[1]hard constraints.

The existing constrained K-means clustering algorithm (Cop-Kmeans) [4] searches for a clustering solution that satisfies all given constraints with the following steps employed: 1. Cluster centroids are calculated based on labels of instances. 2. All instances are relabeled with its nearest feasible cluster under the condition that the assignment does not break any instance-level constraints. If no feasible cluster is available for the assignment of an instance, backtracking and reassigning until a feasible assignment is reached. Constrained K-means clustering algorithm outperforms traditional K-means clustering algorithm without instance-level constraints. However, it at least has three major drawbacks. First, constrained K-means clustering algorithm is a hill climbing search method that can only guarantee a local optimal clustering solution. Second, when the number of instance-level constraints is large, effects of unconstrained instances become weak. The algorithm turns into finding a partition of the data that satisfies all given constraints, but not the one to minimize the predefined clustering criterion [3]. Third, assignments of instances into groups are order-sensitive and backtracking is sometimes very time-consuming [11]. Apart from Cop-Kmeans, another state-of-the-art K-means clustering algorithm with instance-level constraints is the pairwise constrained K-means clustering algorithm (PCKmeans) [8], that uses a greedy search method to optimize the following objective function:

$$\min \sum_{k=1}^{K} \sum_{i=1}^{L} \left[\delta(x_i, C_k) \sum_{j=1}^{N} (x_{ij} - c_{kj})^2 \right] \tag{11}$$

$$+\lambda \times \left[\sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \delta(C, Con(i,j)) \right]$$

where λ is called as the incurred cost that is used to measure how much penalty should be added into the traditional within-cluster variation clustering criterion if a pairwise instance-level constraint is violated. In [8], the value of λ is provided by the user. The optimization process of PCKmeans is very similar to the traditional K-means clustering algorithm, but no backtracking step is required and part of instance-level constraints can be violated in its final clustering solution. Like Cop-kmeans, PCKmeans is only able to guarantee a local optimal clustering solution.

Genetic algorithm is a class of heuristic search algorithms based on the mechanism of nature selection. It has been widely applied into the area of data analysis such as data clustering, feature selection and machine vision. In this paper, we explore the genetic algorithm to optimize the objective function (11). The following section will go into details of describing genetic-guided semi-supervised clustering with instance-level constraints.

4. GENETIC-GUIDED SEMI-SUPERVISED CLUSTERING WITH INSTANCE-LEVEL CONSTRAINTS

This paper adopts a hybrid genetic algorithm to solve the problem of semi-supervised clustering with instance-level constraints. We term this algorithm as the hybrid genetic-guided semi-supervised clustering algorithm with instance-level constraints (Cop-HGA). Like the standard genetic algorithm, Cop-HGA maintains a population of coded candidate clustering solutions during its search. There are several

effective approaches for encoding candidate clustering solutions in the existing literature of genetic-guided clustering algorithms [12]. In this paper, we use the *string-of-group* encoding strategy because of its simplicity and wide applications. In *string-of-group* encoded genetic-guided clustering algorithms, each chromosome in the population is considered as an integer string of length L, where L is the number of instances of being partitioned and each element in the chromosome represents the label of the cluster that an instance is classified into. Let $\{x_1, x_2, ..., x_L\}$ be a set of instances and K is the number of groups that has been known beforehand, a candidate solution can be coded as: $\{I_1, I_2, ..., I_L\}$, where I_j is the label of the cluster into which the instance x_j is partitioned and $I_j \in \{1, 2, ..., K\}$. For example, the chromosome $\{1,\ 2,\ 3\}$ and the chromosome $\{2,\ 2,\ 1\}$ represent candidate clustering solutions $\{\{x_1\}\{x_2\}\{x_3\}\}$ and $\{\{x_1, x_2\}\{x_3\}\}$ respectively.

Cop-HGA uses the formula (11) to calculate fitness values of all candidate clustering solutions in the population. It is noted that candidate solutions with smaller values of the objective function are considered as better candidate clustering solutions. This is because the purpose of Cop-HGA is to find a partition of a set of instances such that the objective (11) is minimized. In formula (11), the value λ is used to balance between the clustering criterion and the instance-level constraints. It was known that different data sets may have different "best" values of λ [8]. In this paper, the setting of the value λ is based on the assumption that clustering criterion and prior background knowledge contribute equally. Therefore, Cop-HGA sets the value of λ as the average value of the square distance from an instance to its cluster centroid. Let M be the number of candidate solutions in the population and $C^{(i)}$ denote the i^{th} candidate clustering solution in the population, then λ is set as:

$$\lambda = \frac{\sum_{i=1}^{M} s(C^{(i)})}{M \times L} \tag{12}$$

Our experimental results indicated that Cop-HGA can always perform well if λ is set as the average value of the square distance from an instance to its cluster centroid. In this paper, Cop-HGA uses the tournament selection operator and its tournament size is fixed to 2. The tournament selection operator works with the following steps employed: two candidate solutions are randomly selected from the population and let them compete, then the one with the smaller within-cluster variation is considered as the winner and selected. The best candidate solution in each generation is directly copied into the new population at the next generation. It is noted that commonly used one-point crossover operator of genetic algorithm is discarded from Cop-HGA. This is because one-point crossover operator frequently reproduces low-quality clustering solutions that violate many instance-level constraints. For example, given two clustering solutions $\{1, 2, 1\}$ and $\{2, 1, 2\}$, and one Must-link constraint that the instance x_1 and the instance x_2 must be classified into the same group. It is known from the definition of Must-link constraints that both clustering solutions satisfy the Must-link constraint. However, their offsprings $\{1, 2, 2\}$ and $\{2, 1, 1\}$ reproduced by the one-point crossover operator will break the Must-link constraint that the instance x_1 and the instance x_2 must be classified into the same group.

A new genetic operator, termed as One-step Constrained K-means operator (OCK), is proposed to speed up the convergence of Cop-HGA. OCK works with the following steps employed: 1. The centroids of clusters are calculated based on labels of instances; 2. Instances are reassigned into its closest feasible groups under the guidance of cluster centroids and instance-level constraints. If no feasible cluster is available for the assignment of an instance, then the instance is assigned into its closest cluster or the cluster such that the number of violated instance-level constraints is minimal. Let I denote a candidate clustering solution in the population and I_j denote the label of the instance x_j to be assigned and $\{x_1, x_2, ..., x_{(j-1)}\}$ be a set of instances that have been assigned before the assignment of the instance x_j. The number of violated instance-level constraints if the instance x_j is assigned into the k^{th} group can be calculated as follows:

$$A(I_j = k) = \sum_{l=1}^{j-1} \delta(I_j, I_l) \tag{13}$$

where

$$\delta(I_j, I_l) = \begin{cases} 1 & \text{if } Con(j,l) = 1 \text{ and } I_l \neq k; \\ 1 & \text{if } Con(j,l) = -1 \text{ and } I_l = k; \\ 0 & \text{if otherwise;} \end{cases} \tag{14}$$

for $k = 1, 2, ..., K$. After the values of $\{A(I_j = k),\ k = 1, ..., K\}$ are calculated, OCK identifies all possible feasible assignments of the instance x_j according to:

$$\mathcal{Q} = \{k | A(I_j = k) = 0,\ k = 1, 2, ..., K\} \tag{15}$$

If $\mathcal{Q}$ is not an empty set, then the instance x_j is assigned into its closest feasible cluster, that is:

$$I_j = \arg\min_{k \in \mathcal{Q}} \|x_j, c_k\|^2 \tag{16}$$

otherwise, the instance x_j is assigned into its closest cluster or the cluster such that the number of violated instance-level constraints is minimal, that is:

$$I_j = \begin{cases} \arg\min_{k=\{1,2,...,K\}} \|x_j, c_k\|^2 & \text{if } rand(1) < a; \\ \arg\min_{k=\{1,2,...,K\}} A(I_j = k) & \text{if otherwise;} \end{cases} \tag{17}$$

where $\{c_1, c_2, ..., c_K\}$ are the centroids of clusters and a is the control parameter that is used to determine which assignment strategy is adopted.

5. EXPERIMENTS

5.1 Experimental settings

Five real data sets from UCI Machine Learning Repository [13] were selected to test the performance of Cop-HGA: Iris data set (150 instances, 4 features, 3 clusters), Glass data set (214 instances, 9 features, 6 clusters), Thyroid data set (215 instances, 5 features, 3 clusters), Wisconsion data set (699 instances, 9 features, 2 clusters) and Pima data set (768 instances, 8 features, 2 clusters). Parameters settings in experiments were set as follows: Population size of Cop-HGA was fixed to 500 for Iris data set, Glass data set, Thyroid data set and 1000 for Wisconsion data set, Pima data set. The standard mutation operator of genetic algorithm was adopted and its mutation rate was set as 0.01. The tournament selection operator was used and its tournament size was fixed to 2. All experiments were independently executed for 30 runs and their average results were reported. Instance-level constraints were generated through randomly selecting two instances from data sets and checking their

labels[2]. If their labels were the same, then a Must-link constraint connecting these two instances was generated; otherwise, a Cannot-link constraint was achieved. The accuracy of a clustering solution was measured by the Rand Index approach as follows [14]:

$$\|I^{(i)}, I^{(acc)}\| = \frac{2 \cdot (n_{00} + n_{11})}{n \cdot (n-1)}$$

where n_{11} is the number of pairs of instances which are both in the same group in $I^{(i)}$ and also both in the same group in $I^{(acc)}$ and n_{00} denotes the number of pairs of instances which are in different groups in $I^{(i)}$ and also in different groups in $I^{(acc)}$ and $I^{(acc)}$ is the accurate partition that was known for all tested UCI data sets.

5.2 Performance of Cop-HGA

First, we fixed the control parameter a to 0.3 and studied clustering accuracies of Cop-HGA under different numbers of instance-level constraints. Experimental results of Cop-HGA were compared with those obtained by two state-of-the-art constrained K-means clustering algorithms: Cop-kmeans [4] and CPKmeans [8] and another two unconstrained K-means clustering algorithms: K-means clustering algorithm and Genetic K-mean algorithm (GKA) [15]. The results were shown in Figure 1. The first phenomenon observed from Figure 1 was that instance-level constraints can significantly improve clustering accuracies of both traditional K-means clustering algorithm and genetic-guided K-means clustering algorithm. Another phenomenon observed from Figure 1 is that Cop-HGA outperforms Cop-kmeans and CPKmeans. For example, if 500 instance-level constraints are added, Cop-kmeans can achieve around 94% accuracy for Iris data set, 80% accuracy for Glass data set, 83% for Thyroid data set, 94% accuracy for Wisconsion data set and 67% accuracy for Pima data set and CPKmeans can achieve 96% accuracy for Iris data set, 75% accuracy for Glass data set, around 77% accuracy of Thyroid data set, 95% accuracy for Wisconsion data set and around 70% accuracy for Pima data set. The results of Cop-HGA are the best for all five data sets. Its clustering accuracies are around 99.5% for Iris data set, 88.0% accuracy for Glass data set, 97.5% for Thyroid data set, 96.2% accuracy for Wisconsion data set and around 75.0% accuracy for Pima data set.

5.3 Effectiveness of OCK operator

To demonstrate the potential of OCK operator, we fixed the number of instance-level constraints to 300, the control parameter a to 0.3 and executed genetic-guided semi-supervised K-means clustering algorithm with instance-level constraints with the following four different genetic operators:

- Cop-HGA: hybrid constrained genetic-guided clustering algorithm with OCK operator;
- Cop-SGA: constrained genetic-guided clustering algorithm with standard mutation operator;
- Cop-CGA: constrained genetic-guided clustering algorithm with one-step constraint satisfaction operator;
- Cop-KGA: constrained genetic-guided clustering algorithm with one-step K-means operator;

[2]For UCI data sets, the accurate labels of instances are available from the data sets.

The results were given in Figures 2-4. It can be observed from Figures 2-4 that the standard mutation operator did not perform well enough for semi-supervised clustering with instance-level constraints. It was unable to effectively reproduce new candidate clustering solutions that minimize the within-cluster variation or the number of unsatisfied instance-level constraints fast. One-step constraint satisfaction operator performed better than the standard mutation operator. It biased the search for clustering solutions that satisfy the given instance-level constraints. Therefore, Cop-CGA quickly found a clustering solution satisfying the given instance-level constraints. However, the one-step constraint satisfaction operator neglected the objective of the within-clustering variation, therefore was not be enough to search for a high-quality clustering solution with a small within-cluster variation. Unlike the one-step constraint satisfaction operator, the clustering solution captured by constrained genetic-guided clustering algorithm with one-step K-means operator (Cop-KGA) usually had a small within-cluster variation, but sometimes a large value of the number of unsatisfied instance-level constraints. This was because the one-step K-means operator benefited for the minimization of the within-cluster variation, but did not benefit for the minimization of the number of unsatisfied instance-level constraints. For all tested data sets, Cop-HGA performed the best. It reproduced high quality clustering solutions that drew a good balance between the clustering criterion and the instance-level constraint fast. The above observations let us know that OCK operator is better than the standard mutation operator, the one-step constraint satisfaction operator and the one-step K-means operator. It can speed up the search of genetic-guided semi-supervised clustering with instance-level constraints for a satisfactory clustering solution that draws a good balance between the clustering criterion and the instance-level background knowledge.

5.4 Effect of the control parameter

Lastly, the effect of the control parameter a on the performance of Cop-HGA was studied. Figure 5 gives the experimental results. It can be observed from Figure 5 that Cop-HGA with different control parameters a may lead to clustering solutions with different accuracies. The value of the control parameter a from 0.3 to 0.5 can always draw a good balance between the clustering criterion and the background knowledge. Therefore, Cop-HGA under this value of the control parameter a often achieved a high-quality clustering solution.

6. CONCLUSION

This paper has considered semi-supervised clustering algorithm with instance-level constraints as a constrained combinatory optimization problem and proposed a novel hybrid genetic-guided semi-supervised clustering algorithm, termed as Cop-HGA. Cop-HGA combines the robustness of genetic algorithm and the rapidity of K-means clustering algorithm for semi-supervised clustering with instance-level constraints. The effectiveness of Cop-HGA and its OCK operator has been confirmed by several real data sets with artificial instance-level constraints.

7. ACKNOWLEDGMENTS

This work was supported by grant 7002073, the Research

Grants Council of the Hong Kong Special Administrative Region, China RGC Ref: 9041236.

8. REFERENCES

[1] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Survey*, 13:264–323, 1999.

[2] K. Wagstaff. *Intelligent Clustering with Instance-Level Constraints*. Department of Computer Science and Engineering, Cornell University, 2002.

[3] M. Law. *Clustering, Dimensionality Reduction, and Side Information*. Department of Computer Science and Engineering, Michigan State University, 2006.

[4] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, pages 577–584, 2001.

[5] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *International Conference on Machine Learning*, pages 1103–1110, 2000.

[6] D. Klein, S. D. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning*, pages 307–314, 2002.

[7] Z. Lu and T.K. Leen. Semi-supervised learning with penalized probabilistic clustering. In *Advances in Neural Information Processing Systems*, 2005.

[8] S. Basu, M. Bilenko, and R.J. Mooney. A probabilistic framework for semi-supervised clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 56–68, 2004.

[9] E.P. Xing, A. Y. Ng, M.I. Jordan, and S. Russell. Distance metrix learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 505–512, 2002.

[10] S. Basu and I. Davidson. Clustering with constraints: Theory and practice. In *ACM KDD2006 Tutorials*, 2006.

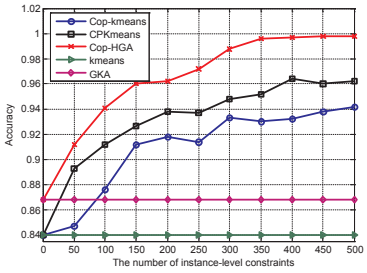
[11] Y. Hong and S. Kwong. Learning assignment order of instances for constrained k-means clustering algorithm. *IEEE Transactions on System, Man and Cybernetics, Part B*, Under Review.

[12] Y. Hong, S. Kwong, H. Xiong, and Qingsheng Ren. Data clustering using virtual population based incremental learning algorithm with similarity matrix encoding strategy. In *GECCO 2008*, to appear.

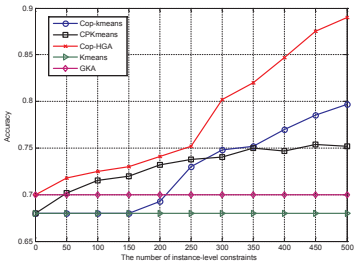
[13] C. Blake and C. Merz. *UCI Machine Learning Repository*. http://www.ics.uci.edu/mlearn/MLRepository.html, 1998.

[14] W.M. Rand. Objective criterion for the evaluation of clustering methods. *Journal of Americal Statistical Association*, 66:846–850, 1970.

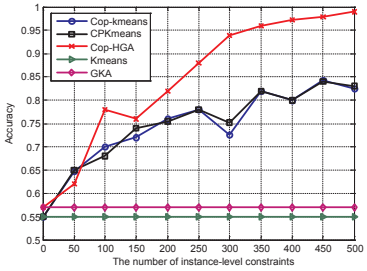
[15] K. Krishna and M. Murty. Genetic k-means algorithm. *IEEE Transactions on System, Man, and Cybernetics-Part B*, 29:433–439, 1999.



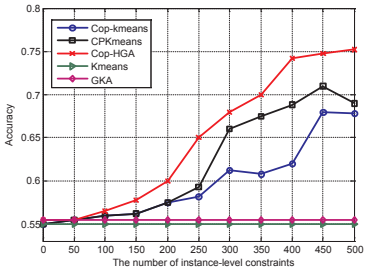
(1) Iris data set



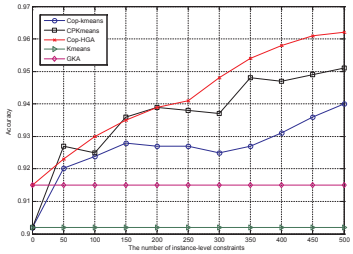
(2) Glass data set



(3) Thyroid data set

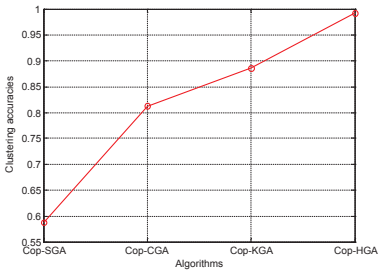


(4) Pima data set

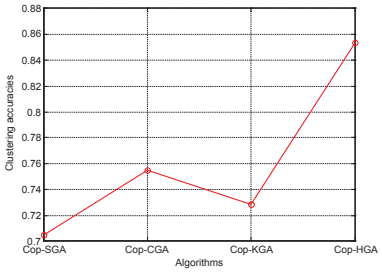


(5) Wisconsion data set

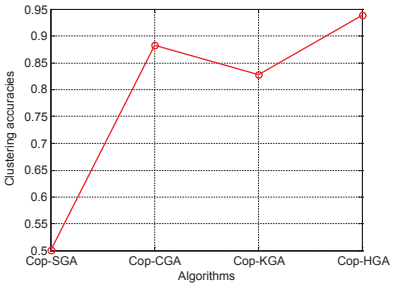
Figure 1: Comparisons between Cop-HGA, constrained and unconstrained clustering algorithms.



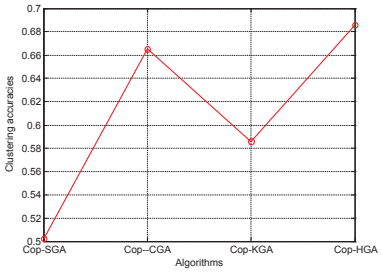
(1) Iris data set



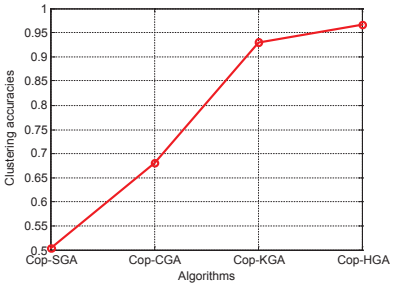
(2) Glass data set



(3) Thyroid data set

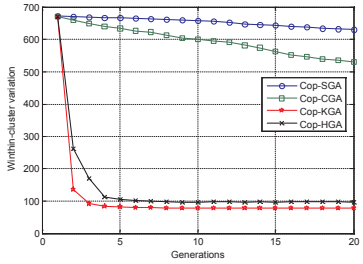


(4) Pima data set

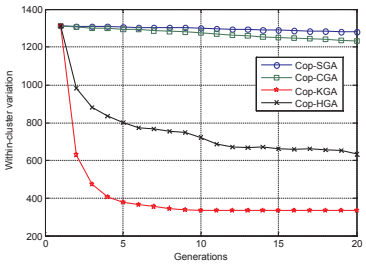


(5) Wisconsion data set

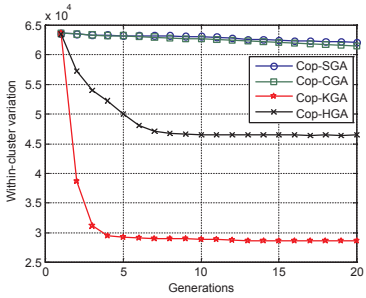
Figure 2: Accuracies obtained by different genetic-guided semi-supervised clusterings algorithms.



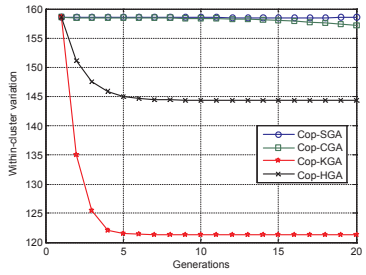
(1) Iris data set



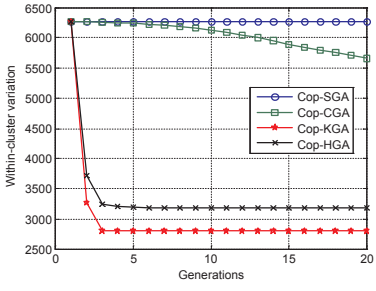
(2) Glass data set



(3) Thyroid data set

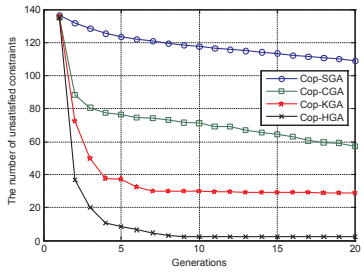


(4) Pima data set

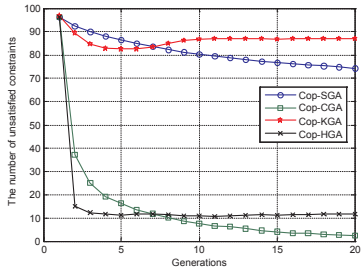


(5) Wisconsion data set

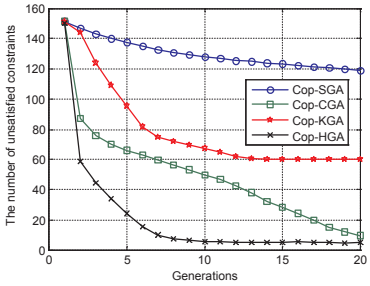
Figure 3: Within-cluster variations obtained by different genetic algorithms.



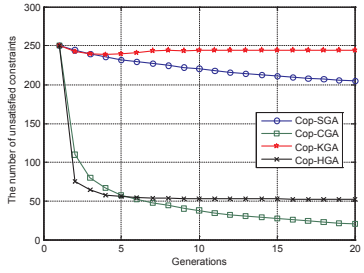
(1) Iris data set



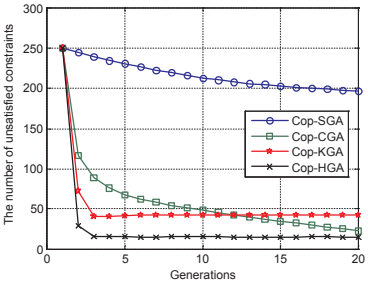
(2) Glass data set



(3) Thyroid data set

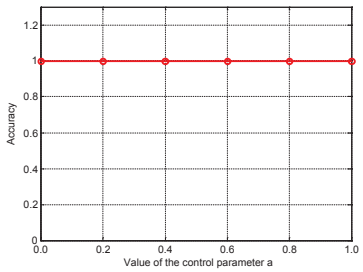


(4) Pima data set

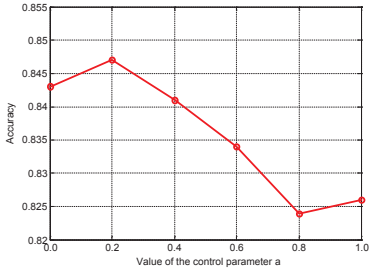


(5) Wisconsion data set

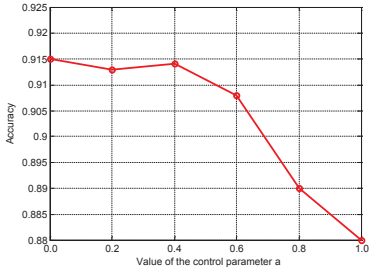
Figure 4: The numbers of unsatisfied constraints obtained by different genetic algorithms.



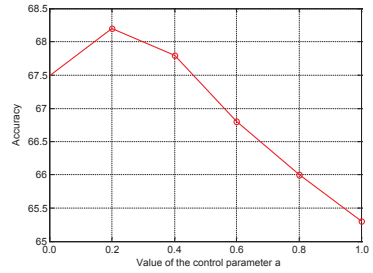
(1) Iris data set



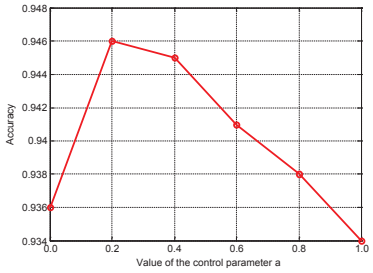
(2) Glass data set



(3) Thyroid data set



(4) Pima data set



(5) Wisconsion data set

Figure 5: Clustering accuracies obtained by Cop-HGA with different control parameters a.