Leader Ant Clustering with Constraints

Viet-Vu Vu, Nicolas Labroche, Bernadette Bouchon-Meunier
Université Pierre et Marie Curie - Paris 6, CNRS UMR 7606, LIP6
104 avenue du Président Kennedy, F-75016 Paris, France
Email: {viet-vu.vu, nicolas.labroche, bernadette.bouchon-meunier}@lip6.fr

Abstract—In recent years, clustering with constraints has become a topic of significant interest for many researchers because it allows to take into account the knowledge from the domain, expressed as a set of constraints, and thus to improve the efficiency of the analysis. For example, these approaches can take place in an interactive process where a user iteratively expresses new constraints to refine previous clustering results. In this paper, we propose three new variants of the Leader Ant Clustering with Constraint algorithm (MCLA, MELA and CELA) that implements the following constraints: the must-link, cannot-link constraints and \( \varepsilon \)-constraints. These algorithms have been compared to other constraint based clustering algorithms such as K-Means clustering with constraints and the original Leader Ant clustering algorithm. Our experiments show that, on UCI machine learning and artificial data sets, our approach compares well to the other algorithms.

Keywords - clustering; constraint; artificial ants.

1. INTRODUCTION

In recent years, clustering with constraints has become a topic of significant interest for many researchers because these methods allow to take into account the knowledge from the domain, expressed as a set of constraints, and thus to improve the quality of the analysis. These approaches can take place in an interactive process where a user iteratively expresses new constraints to refine previous clustering results. Constraints can be particularly beneficial in data mining, when precise definitions of classes are absent. The aspects of algorithms, theory and applications of the problem of clustering with constraints have been introduced in [1].

The constraints are indeed much easier to provide than class labels even when the user has a very incomplete prior knowledge of the domain. In some applications, the constraints can be collected automatically in a preprocessing step of the learning algorithm [18].

Let \( S = \{s_1, s_2,...,s_n\} \) be a set of \( n \) data points which are to be partitioned into clusters. Two types of constraints are proposed by Wagstaff et al. in [16]:

- **Must-link** constraints specify that two points \( s_i \) and \( s_j \) \( (i \neq j) \) in \( S \) have to be in the same cluster.

- **Cannot-link** constraints specify that two point \( s_i \) and \( s_j \) \( (i \neq j) \) in \( S \) must not be placed in the same cluster.

For any pair of points \( s_i \) and \( s_j \) in \( S \), the distance between them is denoted by \( d(s_i, s_j) \) with a symmetric property so that \( d(s_i, s_j) = d(s_j, s_i) \). Davidson et al. proposed two other constraints [4]:

- **\( \delta \)-Constraint**: this constraint specifies a value \( \delta > 0 \). Formally, for any pair of clusters \( S_i \) and \( S_j \) \( (i \neq j) \), and any pair of points \( s_p \) and \( s_q \) such that \( s_p \in S_i \) and \( s_q \in S_j \), \( d(s_p, s_q) \geq \delta \).

- **\( \varepsilon \)-Constraint**: this constraint specifies a value \( \varepsilon > 0 \) and the feasibility requirement is the following: for any cluster \( S_i \) containing two or more points and for any point \( s_p \in S_i \), there must exist another point \( s_q \in S_i \) such that \( d(s_p, s_q) \leq \varepsilon \).

Figure 1 and figure 2 illustrates different types of prior knowledge that can be included in the process of classifying data: dots correspond to points without any labels; points with labels are denoted by circles, asterisks and crosses. In figure 2 (left), the must-link and cannot-link constraints are denoted by solid and dashed lines [24].

![Figure 1. Spectrum of supervised (left) and partially labelled (right) learning](image1)

![Figure 2. Spectrum of constrained (left) and unsupervised (right) learning](image2)

We can divide previous work on clustering under constraints into two main families [4, 7, 25]: 1) where the constraints help the algorithm to learn a distance/objective
function and 2) where the constraints are used as “hints” to guide the algorithm to a useful solution.

Ant-based clustering was first introduced by Deneubourg et al. [22] to explain different types of naturally-occurring emergent phenomena. Ant-based clustering algorithms are optimization methods that rely on the modeling of collective behaviors of real ants. When clustering, ants gather items to form heaps; an example of this being the cemetery formation (i.e., the clustering of dead corpses) observed in the species *Pheidole pallidula*.

In [8, 9], Labroche proposed an ant-based approach that is inspired by the chemical recognition system of ants. In this paper, we introduce three new methods named Leader with Constraint algorithm (MCLA, MELA and CELA) that derive from the Leader Ant algorithm and that introduce the following constraints: the must-link, cannot-link constraints and -constraints. These algorithms have been compared to other constraint based clustering algorithms such as K-Means clustering with constraints and the original Leader Ant clustering algorithm. Our experiments show that, on UCI machine learning and artificial data sets, our approach compares well to the other algorithms.

The rest of the paper is organized as follows: Section II discusses the related work. We present the Leader Ant with Constraints algorithms in section III. Section IV contains our experimentations. Finally, section V summarizes our results and proposes some directions for future work.

II. RELATED WORD

Many researches have been conducted on learning methods with constraints: partitioning algorithm [4, 16], hierarchical algorithm [5], EM clustering with pairwise constraints [1, 23, 27], density-based algorithm [15], incremental constrained clustering [3], Support Vector Machine clustering with constraints [20] and co-clustering with constraints [12, 13].

Many applications of constraint-based clustering include the detection of road lanes from GPS data [17], helping the navigation of Sony Aibo Robot [4], Noun Phrase Coreference Resolution and Spectral Analysis [17], video object classification [18], and interactive visual clustering [1].

The last five years, constraints have been used in the classification problem to improve results such as in classification with pairwise constraints using SVM [11] and in discriminative learning framework with pairwise constraints for video object classification [18].

In [4], Davidson et al. proposed the K-means with constraints (KMC). The key idea of KMC is to use the constraints to create a new differentiable error function called the constrained vector quantization error that reduces the number of iterations until convergence. The results showed that KMC converged on average with 25% less iterations while satisfying the vast majority of constraints.

In [16], Wagstaff et al. proposed a K-Means algorithm with must-link and cannot-link constraints named COP-KMeans (Table I). The COP-KMeans attempts to find a partition that minimizes the vector quantization error and also satisfies all the constraints. Wagstaff’s empirical results show that using constraints can improve the performance of clustering [17].
III. LEADER ANT CLUSTERING WITH CONSTRAINTS

As the three new methods proposed in this paper relies on the Leader Ant Clustering algorithm, we first recall in subsection A the main principles of this approach. Second we describe in subsection B, C and D the new variants and the introduction of the three constraints - must-link, cannot-link and ε-constraints - in the clustering process.

A. The Leader Ant clustering algorithm

The Leader Ant clustering algorithm (LA) is inspired from the chemical recognition system of ants. The following subsections first detail the biological system and then describe the model that has been developed more specifically to solve the clustering problem.

1) The Chemical Recognition System of Ants

In the biological system, each ant possesses its own odor called label that is spread over its cuticle (its “skin”). This label acts as an identity card and is partially determined by the genome of the ant and by the substances extracted from its environment (mainly the nest materials and the food). During their youth, ants learn to distinguish the labels of the colony members and learn a neuronal template of what should be the label of a nestmate. This template is continually updated and is used at each meeting between two ants, to decide if they should accept each other and exchange chemical cues (by trophallaxies, allo-grooming or simple social contacts). The continuous chemical exchanges between the nestmates lead to the establishment of a colonial odor that is shared and recognized by every nestmate s, according to the “Gestalt theory” [6].

2) The Leader Ant Model

The Leader Ant model (LA) is inspired from the real ants system, adapted to solve the clustering problem and for performance purposes. In LA, an artificial ant is described by three parameters.

- The genome is associated with an unique object of the data set;
- The template is the same for all artificial ants and is either fixed or computed experimentally as the mean value of the distance values \(d(i, j)\) estimated between \(N_{\text{learn}}\) couples of ants \(i\) and \(j\) randomly selected.

\[
\text{template} = \frac{\sum \limits_{i}^{N_{\text{learn}}} \sum \limits_{j}^{N_{\text{learn}}} d(i, j)}{N_{\text{learn}}^2} \tag{1}
\]

with \(d(i, j)\), the distance value between the object associated to the ant \(i\) and the object of the ant \(j\).

- The label reflects the nest membership of each artificial ant. At the beginning, this value is set to zero as no hypothesis is made concerning the initial membership of ants.

LA is a one-pass agglomerative algorithm that iteratively selects at random a new ant \(a\) (that has not been already assigned to a nest), and determines its label or nest membership by simulating \(N_{\text{meetings}}\) meetings with randomly selected ants from each existing nest \(k\) in \([0, N_{\text{MaxNests}}]\). During these meetings, the ant \(a\) estimates the similarity of its genome with those of ants from the evaluated nest \(k\). At the end, the distance \(D(a, k)\) between the ant \(a\) and the nest \(k\) is computed as the mean distance over the \(N_{\text{meetings}}\) meetings.

\[
D(a, k) = \frac{1}{N_{\text{meetings}}} \sum \limits_{j=1}^{N_{\text{meetings}}} d(a, \text{ant}_j^k) \tag{2}
\]

where \(\text{ant}_j^k\) is the \(j^{th}\), \(j \in [1, N_{\text{meetings}}]\), randomly selected ant from nest \(k\). If no nest exists or if the mean distance value is under the template value, the ant creates its own new nest.

\[
N_{\text{MaxNests}} = N_{\text{MaxNests}} + 1 \text{ (create a new nest)}
\]

In the opposite case, the ant joins the nest with the lowest mean distance value by setting its label as follows:

\[
\text{Label}_a = \text{Label}_k
\]

Finally, when all ants are assigned to a nest, the smallest nests whose size is under a fixed percentage of the total number of objects \(n\) can optionally be deleted and their ants reassigned to the other clusters.

The LA algorithm runs in linear time with the number of points of data set \(n\) and the number of clusters \(k\), i.e. \(O(kn)\) as each point is compared to each existing cluster to decide its assignment. Table II presents the algorithm of the Leader Ant method.

<table>
<thead>
<tr>
<th>TABLE II. THE LA ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> data set (S) with (n) points</td>
</tr>
<tr>
<td><strong>Output:</strong> partition of the data set</td>
</tr>
<tr>
<td>1. Initialization of Artificial Ants</td>
</tr>
<tr>
<td>2. Template computation or template fixed experimentally</td>
</tr>
<tr>
<td>3. Iterative nests Building</td>
</tr>
<tr>
<td>3.1 Selection of an artificial ant (a)</td>
</tr>
<tr>
<td>3.2 For each existing nest (k)</td>
</tr>
<tr>
<td>- Random meetings with ants from (k)</td>
</tr>
<tr>
<td>- Estimation of distance between (a) and nest (k)</td>
</tr>
<tr>
<td>4. Deletion of smallest Nests (option)</td>
</tr>
</tbody>
</table>

B. Leader Ant clustering with Must-link and Cannot-link constraints (MCLA)

We now present the MCLA algorithm which integrates the must-link and cannot-link constraints to LA algorithm. We use the must-link and cannot-link constraints as “hints” to guide the algorithm to a useful solution.

As it is well known, must-link constraints are transitive; that is to say, must-link constraints \((s_a, s_b)\) and \((s_b, s_c)\) imply that there exists a must-link constraint \((s_a, s_c)\). Thus, the two constraints can be combined into a single must-link constraint, namely \((s_a, s_b, s_c)\). So, a given collection \(\text{Con}_a\) of must-link constraints can be transformed into an equivalent collection \(M = \{M_1, M_2, ..., M_r\}\) of constraints, by computing the transitive closure of \(\text{Con}_a\) [4].
The transitive closure computation in the algorithm can be carried out as follows. Construct an undirected graph $G$ with $n$ node, one node for each point in the data set, and an edge between two nodes if the corresponding points appear together in the must-link constraint (figure 3). Then, the connected components of $G$ give the sets of objects in the transitive closure. Therefore, the connected components can be found in $O(n+m)$ time [2], where $m$ is the number of must-link constraints (the edges of $G$). We use the set of transitive closure to reduce the number of points which has to be considered during the clustering.

The set of cannot-link can be used directly at each iteration to determine the clusters when estimating the distance between a nest and an artificial ant (equation 2). We present MCLA algorithm in figure 4.
C. Leader Ant clustering with Must-link and \( \varepsilon \)-constraints (MELA)

In this section we propose the MELA algorithm (see figure 4) that implements the must-link constraints and \( \varepsilon \)-constraint in LA algorithm. Similarly to MCLA algorithm, we construct the transitive closure for the must-link constraints (figure 6). It is important to notice that the transitive closure has to satisfy the \( \varepsilon \)-constraint: for any transitive closure \( M_k \) and for any point \( s_p \in M_k \), there must be another point \( s_q \in M_k \) such that \( d(s_p, s_q) \leq \varepsilon \). If not, we have to find some points of the data set to complement the transitive closures. Figure 6 show that there are some points which relate to the transitive closure (by dashed line) and allow the \( \varepsilon \)-constraint to be satisfied. If there exists a transitive closure which cannot find the points to satisfy the \( \varepsilon \)-constraint, the algorithm will stop and the output is “No solution”.

In the MELA algorithm, we use a R*-Tree structure to find the nearest neighbor points for each point in the data set [21]. With R*-Tree, the time to perform the queries to find the nearest neighbor points is \( O(\log(n)) \) (\( n \) is being the number of objects in the data set). We present the MELA algorithm in figure 4.

![Figure 6. Construction the transitive closure of the Must-link (ML) for the MELA Algorithm](image)

D. Leader Ant clustering with Cannot-link and \( \varepsilon \)-constraints (CELA)

In this section we derive a constraint version of LA algorithm by using cannot-link constraints and \( \varepsilon \)-constraint. We use the cannot-link constraints and \( \varepsilon \)-constraints directly in the clustering process. As for MCLA, the set of cannot-link constraints can be used directly at each iteration to find the clusters when estimating the distance between a nest and an artificial ant (equation 2).

Similarly to the MELA algorithm, we use an R*-Tree structure to determine the nearest neighbors of each points. We present the CELA algorithm in figure 5.

IV. EXPERIMENTAL RESULTS

In this section, we report experiments to evaluate the efficiency of our algorithm. We compare the results of MCLA with the COP-KMeans algorithm as proposed by Wagstaff et al. in [19]. We also show to which extent the number of constraints influences the clustering of the CELA and MELA algorithm compared to the original LA algorithm.

A. The data sets

We use real data sets from the Machine Learning repository named Glass, Iris, Ionosphere, Wine, Thyroid and Soybean. We also used 6 artificial data sets named Art\(_i\), \( 1 \leq i \leq 6 \). They have been generated according to Gaussian or uniform laws with distinct difficulties [10]. Tables III and IV present the number of points \( n \), the number of attributes \( m \) and the number of clusters \( k \) in the partition of each data set.

<table>
<thead>
<tr>
<th>Files</th>
<th>( n )</th>
<th>( m )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Files</th>
<th>( n )</th>
<th>( m )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thyroid</td>
<td>215</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Soybean</td>
<td>47</td>
<td>35</td>
<td>4</td>
</tr>
<tr>
<td>Art1</td>
<td>400</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Art2</td>
<td>1000</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Art3</td>
<td>1100</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Art4</td>
<td>200</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Art5</td>
<td>900</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>Art6</td>
<td>400</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

B. Evaluation method

The data set used for the evaluation includes a “correct answer” or label for each data point. We use the labels in a post-processing step for evaluating the performance of our approaches.

To evaluate the similarity between the expected partition and the partition produced by our method, we use the Rand Index [14]. This measure is based on \( \left( \frac{n(n-1)}{2} \right) \) pairwise comparisons between the \( n \) points of a data set \( D \). For each pair of points \( x_i \) and \( x_j \) in \( D \), a partition assigns them either to the same cluster or to different clusters. A total agreement can then be calculated using:

\[
\text{Rand}(P_1, P_2) = \frac{a + b}{n \ast \left( n - 1 \right) / 2}
\]

We used this measure to evaluate the performance of our approach in all of our experiments.

C. Results

1) Comparison between MCLA and COP-KMeans

In this section, we present the evaluation of MCLA against the COP-KMeans algorithm as it implements the same
constraints (that is to say must-link and cannot-link constraints). We used the same experimental settings as in [19] and thus we only consider four data sets from the Machine Learning repository [26]: Glass, Iris, Ionosphere, Wine (see table III).

The constraints were generated as follows: for each constraint, we randomly picked two points from the data set and checked their labels. If they have the same label, we generate a must-link constraint. Otherwise, we generate a cannot-link constraint. We used 50 constraints of must-link and cannot-link as in [19], then we conducted 100 runs for each data set and we averaged the results.

We compared MCLA and COP-KMeans on three aspects: average minimum accuracy, average mean accuracy and average maximum accuracy over 100 runs. The results are showed in figure 7.

As can be seen from the results on Glass, Iris and Wine data sets, MCLA shows better accuracy and convergence than COP-KMeans while it does not need the number of clusters K as input.

With the Ionosphere data set, MCLA performs slightly better for maximum accuracy but is worse with regard to minimum accuracy. The results can be explained by the fact that in some cases MCLA under-estimates or over-estimates the number of clusters and is thus penalized compared to the COP-KMeans by the Rand distance that is sensitive to difference between the number of theoretical clusters and the number in the output partition.

An other aspect that can explain the differences between the observed results is that the algorithms COP-KMeans and MCLA do not share exactly the same objective: COP-KMeans aims at satisfying all the constraints whereas MCLA uses the constraints to refine the search space and to indirectly help to define the number of clusters as such information is not initially available as a parameter of the method.

2) **Comparison of MELA and CELA with LA**

Now, we compare MELA and CELA with LA algorithm with some data sets (see Table IV). Each graph demonstrates the change in accuracy as more constraints are made available to the algorithm. We then conducted 20 runs for each data set and averaged the results. The must-link and cannot-link constraints are generated as in subsection 1. The ε-constraint is chosen experimentally by user.

At first with the thyroid data set, without any constraints, the LA algorithm achieves an accuracy of 82.3% (see figure 8). Overall accuracy steadily increases with incorporation of constraints. The accuracy reaches 86.5% with MELA and 90.2% with CELA after 25 random constraints.

With soybean data set, without any constraints, the LA algorithm achieves an accuracy of 83.3%, but when we used 12 constraints, the results show that the accuracy rises from 83.3% to 87% with both MELA and CELA (figure 9).

Figure 10 presents the results of art data set after we used 5, 10, 15 and 20 constraints. The results improved considerably. From 89.2% without any constrains to 90.5% with MELA algorithm and 91.1% with CELA algorithm.
Figure 8. Thyroid data set

Figure 9. Soybean data set

Figure 10. Art1 data set

Figure 11 shows the results obtained by the Art\(_i\) (2 \(\leq\) i \(\leq\) 6) data sets. In this test, we used the percent of constraints between 2% and 3% to compare MELA, CELA with LA. Although the number of constraints is very small, the accuracy is considerably improved.

Finally, the results also show that the efficiency of algorithms using cannot-link constraints is better than using must-link constraints in general. It can be explained by the fact that cannot-link constraints improve the clustering results when clusters are not well separated in their definition space. In our experiments, the must-link constraints mainly help the algorithm to converge faster as many data points are added to a cluster at the same time.

V. CONCLUSION AND FUTURE WORKS

In this paper, we have presented three new constraint-based clustering algorithms named MCLA, MELA and CELA that rely on the Leader Ant clustering algorithm. We integrated the must-link, cannot-link constraints and \(\varepsilon\)-constraints to improve the quality of clustering.

MCLA algorithm compares well to the COP-KMeans algorithm with the same constraints and using data sets from UCI machine learning. We also compared MELA and CELA with original Leader Ant algorithm. Our experimental results confirm the efficiency of the proposed algorithms.

There are many directions for future research:

- We will develop a version of LA algorithm that uses all of three types of constraints: must-link, cannot-link and \(\varepsilon\)-constraint.
- We will research incremental constraints-clustering which is the main topic of interest.
- A persistent underlying question is the issue of where constraint information comes from, how it can be collected, and how much it should be trusted. We need more research for visualization, interpretation and evaluation of the data set.
- Finally, “When is constrained clustering beneficial, and why?” is the question of interest. Some properties of constraints are proposed by Wagstaff et al. [19], but we need more research to identify the meaningful constraint set properties for practitioners and researchers.
REFERENCES


